



Model-driven engineering for mobile robotic systems: a systematic mapping study

Downloaded from: <https://research.chalmers.se>, 2023-05-06 01:11 UTC

Citation for the original published paper (version of record):

Casalaro, G., Cattivera, G., Ciccozzi, F. et al (2022). Model-driven engineering for mobile robotic systems: a systematic mapping study. *Software and Systems Modeling*, 21(1): 19-49.
<http://dx.doi.org/10.1007/s10270-021-00908-8>

N.B. When citing this work, cite the original published paper.



Model-driven engineering for mobile robotic systems: a systematic mapping study

Giuseppina Lucia Casalaro¹ · Giulio Cattivera¹ · Federico Ciccozzi² · Ivano Malavolta³ · Andreas Wortmann⁴ · Patrizio Pelliccione^{5,6}

Received: 23 June 2020 / Revised: 15 June 2021 / Accepted: 25 June 2021
© The Author(s) 2021

Abstract

Mobile robots operate in various environments (e.g. aquatic, aerial, or terrestrial), they come in many diverse shapes and they are increasingly becoming parts of our lives. The successful engineering of mobile robotics systems demands the interdisciplinary collaboration of experts from different domains, such as mechanical and electrical engineering, artificial intelligence, and systems engineering. Research and industry have tried to tackle this heterogeneity by proposing a multitude of model-driven solutions to engineer the software of mobile robotics systems. However, there is no systematic study of the state of the art in model-driven engineering (MDE) for mobile robotics systems that could guide research or practitioners in finding model-driven solutions and tools to efficiently engineer mobile robotics systems. The paper is contributing to this direction by providing a map of software engineering research in MDE that investigates (1) which types of robots are supported by existing MDE approaches, (2) the types and characteristics of MRSs that are engineered using MDE approaches, (3) a description of how MDE approaches support the engineering of MRSs, (4) how existing MDE approaches are validated, and (5) how tools support existing MDE approaches. We also provide a replication package to assess, extend, and/or replicate the study. The results of this work and the highlighted challenges can guide researchers and practitioners from robotics and software engineering through the research landscape.

Keywords Model-driven engineering · Mobile robot systems · Systematic mapping study

1 Introduction

The term “Robot” was coined by the Czech writer Karel Čapek in 1921 in a science-fiction play. One century later, much of this science-fiction has become reality. Nowadays, a robot might be defined as “an intelligent device with a certain degree of autonomy that contains sensors, control systems, manipulators, power supplies and software all working together to perform the required tasks” [1]. In this paper, we focus on mobile robots, which are robotic systems carried around by locomotive elements and which can operate in various environments (aquatic, aerial, or terrestrial). The operational environment decides the locomotive elements.

Communicated by Juergen Dingel.

✉ Federico Ciccozzi
federico.ciccozzi@mdh.se

Giuseppina Lucia Casalaro
giuseppina.casalaro@univaq.it

Giulio Cattivera
giulio.cattivera@univaq.it

Ivano Malavolta
i.malavolta@vu.nl

Andreas Wortmann
wortmann@se-rwth.de

Patrizio Pelliccione
patrizio.pelliccione@univaq.it

¹ Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila, L'Aquila, Italy

² School of Innovation, Design and Engineering, Mälardalen University, Västerås, Sweden

³ Department of Computer Science, Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

⁴ Chair of Software Engineering, RWTH Aachen University, Aachen, Germany

⁵ Chalmers | University of Gothenburg, Gothenburg, Sweden

⁶ University of L'Aquila, L'Aquila, Italy

Robotics comes in many diverse shapes, which are increasingly becoming part of our lives: automated vehicles bring us to work [2] where we cooperate with industrial robots [3] or service robots [4].

Despite robots becoming ubiquitous, robotics is one of the most challenging fields of cyber-physical systems' research and engineering. The H2020 Multi-Annual Robotics Roadmap ICT-2016¹ describes the production of software for robotic systems as a "craftsmanship in building robotic systems instead of following established engineering processes". On the same line, a previous study identifies one of the main challenges of robotics as promoting "a shift towards well-defined engineering approaches able to stimulate component supply-chains and significantly impact the robotics market-place" [1].

The successful engineering of non-trivial robotics systems demands the interdisciplinary collaboration of experts from different domains, such as mechanical and electrical engineering, artificial intelligence, manipulation, Human-Robot Interaction (HRI), and systems engineering. In all of these domains, experts traditionally employ various kinds of models to describe or prescribe parts of their views on the robots under development. For instance, in mechanical engineering, these models may be CAD geometries or differential equations, in path planning, these may be the environment geometries, in artificial intelligence, these may be artificial neural networks, in HRI these may be user interface models, and in systems engineering, these may be SysML [5] diagrams.

Due to this heterogeneity research and industry have brought forth a multitude of model-driven solutions to engineering the software of Mobile Robot Systems (MRS) [6], such as modelling tools and languages supporting the development of robotics solutions focusing on kinematics, dynamics, sensing, motion planning, force control, reasoning, and architectures. These solutions range from very general modelling techniques, including architecture modelling [7] (e.g. SmartMDS [8] or Orocos [9]) or mission planning [10,11], to specific application domains, such as manufacturing [12], to solutions focusing on individual platforms [13]. This wealth of platforms, application domains, modelling and techniques warrants investigating the state-of-the-art systematically to support researchers and practitioners in identifying and selecting model-driven technologies to facilitating engineering of mobile robot systems.

Although the use of model-driven engineering (MDE) in robotics is deemed promising [14], there is no systematic evidence on which kinds of robots are supported by existing MDE approaches, which type of support models provide,

how existing MDE approaches are validated, and how tools support existing MDE approaches. To better understand all these aspects, this paper provides a map of *software engineering* research in MDE for mobile robotic systems, which can guide researchers and practitioners from robotics and software engineering through the research landscape and help to shed light on future challenges. It is important to highlight that in this work, we explicitly focus on software aspects, and we do not dig into the interplay of software and hardware. This would require a separate study and most probably it would demand dedicated (MDE) approaches and tools, software aspects and specifically in software engineering for robotics [15]. This scopes the work and helps us in identifying the papers to be included in this mapping study. The side effect of this choice would be not providing a deep analysis of the interplay of software and hardware, which would require a separate study. For example, the report on the DARPA Urban Challenge presented in [16] is potentially interesting since it describes the model-based design of an autonomous ground vehicle, but it focusses primarily on systems/control models, rather than software. Also, in this work we do not focus on what needs to be done during runtime in order to guarantee important properties, like reliability, availability, maintainability and safety, as for instance discussed in [17]. It is also important to point out that by construction, this study only considers published papers that are captured by our search and selection process. This would imply that potentially interesting studies and approaches might be left out of the selection, like the ESA TASTE toolchain,² and 20-Sim, an MDE toolchain for modelling, verification, code generation, and simulation of dynamic systems including mobile robots.³

In this work, we aim at answering the following research questions:

- RQ1: What are the publication trends of research in MDE for MRSs?
- RQ2: Which types of MRS are supported by existing MDE approaches?
- RQ3: How do existing MDE approaches support the engineering of MRSs?
- RQ4: How are researchers validating existing MDE approaches for MRSs?
- RQ5: What does the landscape of tools supporting MDE for MRSs look like?

To answer these research questions, we performed a systematic mapping study [18,19]. A mapping study research

¹ H2020 Multi-Annual Robotics Roadmap ICT-2016 <http://sparc-robotics.eu/wp-content/uploads/2014/05/H2020-Robotics-Multi-Annual-Roadmap-ICT-2016.pdf>, 2016.

² <https://taste.tools/>.

³ <https://www.20sim.com/>.

methodology permits, in fact, to identify, classify, and evaluate focus, trends, and open challenges.

The main **contributions** of this paper are:

1. An overview of the academic venues and forums where the community disseminate the research ideas and results in the field, together with the density of scientific interest in MDE for MRSs.
2. The types and characteristics of MRSs engineered using MDE approaches.
3. An overview on how MDE approaches support the engineering of MRSs, especially focusing on employed modelling languages and model transformations. Moreover, we describe the aspects of MRSs that are engineered by MDE approaches, such as behaviour, movement, image recognition, and pathfinding.
4. A precise characterization of the types of performed validations, as well as the characteristics of MRSs used for the evaluation of MDE approaches for MRS.
5. A description of the tools and tool types that support MDE approaches for MRSs.
6. A *replication package* for independently assessing and replicating the study.

Outline of the paper. The remainder of the paper is structured as follows. Section 2 details the background of our research before, Sect. 3 details the design of our study. Afterwards, Sect. 4 presents its results, and Sect. 5 uncovers relations between these. Section 6 discusses results, relations, and their implications. Subsequently, Sect. 7 debates threats to validity, and Sect. 8 highlights related work. Section 9 closes the paper.

2 Model-driven engineering and robotics

In the last couple of decades, the amount and the importance of software and software-intensive systems have increased significantly and their complexity has increased exponentially. The importance of software has increased exponentially along with the complexity of software-intensive systems. Thus, software engineers have begun to leverage models to reduce the conceptual gap [20] that arises from using different abstractions in the problem domains at hand, such as robotics, rather than in the solution domain.

Robotics is a domain driven by four challenges [21]:

1. lack of standardization,
2. its multi-disciplinary nature complicating reuse,
3. uncertain environments, and
4. the lack of sufficiently expressive, domain-specific languages.

Consequently, robotics has motivated software engineering research for decades and many frameworks and toolchains to mitigate the challenges in robotics have been developed. Many approaches to software engineering for robotics aim to improve encapsulation and reuse of robotics software solutions by applying component-based software engineering [22] to robotics [23]. For instance, Orca [24], OROCOS [9], and the Robot Operating System (ROS) [25] contribute solutions to modularize and componentize robotics software architectures. Similarly, the trend towards distributed, cloud-based systems has affected robotics at well [26]. To foster creation of distributed multi-robotics systems that interact through the cloud, various methods [27], platforms [28], and tools have been developed [29].

In line with the fourth challenge, MDE has been identified [30] as a core technology to support developers when designing robotic systems. Consequently, robotics has become a domain of original research on model-driven engineering [31].

Model-driven engineering [31–33] is an umbrella term for software development methodologies that leverage (domain-specific) models as primary development artefacts to reduce this gap. To support efficient engineering with such models, these must be machine-processable, which demands corresponding notations, e.g. in the form of meta-models [34,35] or grammars [36] to describe corresponding domain-specific languages (DSLs).

On one hand, models corresponding to different domain-specific notations can be created, edited, and understood, by experts of the different domains contributing to the development of modern software-intensive systems. On the other hand, such models can be processed by sophisticated modelling tools and infrastructures provided by software engineering experts to perform more complex analyses and transformations automatically—thus liberating domain experts from the need to become software engineering experts themselves. These different types of models can be identified as descriptive models, those mainly used for documentation, and prescriptive models, those mostly used for development [37].

In the last years, several papers have been published presenting how industries use MDE and, therefore discussing the importance of modelling. For instance, the works by Kuhn et al. [38] and Aranda et al. [39] report about the use of MDE at General Motors and the work in [40] describes the experience of introducing MDE at Motorola.

These studies show somehow contradictory findings. The work by Liebel et al. [41] shows that modelling can be suitable for embedded systems, and, given that one understands the consequences of adopting modelling, the study by Vallecillo et al. [42] convinces us that modelling is ready for industry. A different point of view is suggested by other stud-

ies. The work by Gorschek et al. [43] shows that there is a lack of use of modelling in industry. Similar findings can be found in the work by Petre et al. [44]. The work by Heldal et al. [37] points out that in order to understand these contradictory messages, it is important to distinguish between descriptive models and prescriptive models. In fact, this study shows that there are substantial differences in the use, importance during the development, maintenance, and management of these two types of models.

The work by Ciccozzi et al. [45] focuses on mission-critical Internet of Things systems, which are exposed to uncertainty and variability and which require high availability, reliability, safety, security, regulatory compliance, scalability, and serviceability. The work discusses how MDE can help meet the technical challenges. The work by Wortmann et al. [46] focuses on modelling in the smart factory of Industry 4.0 and identified various modelling techniques being used that are tailored specifically to that domain. Successful applications of MDE have been reported in the context of software engineering for various domains [31,47,48].

Model-driven engineering for robotics comes in diverse shapes and guises. For instance, several works have been proposed in the last years on the use of architecture modelling languages and techniques for robotic software systems [49–51]. The research reported in [49] contributes the SmartSoft modelling language for software architectures and behaviour of robotic systems, which is supported by a fully automated code generation toolchain to produce executable robotics applications. The BRICS component model presented in [51] also aims to ease modelling of robotics software architectures by reusing tried and tested component models as black boxes. The work presented in [52] combines multiple modelling languages to describe robot capabilities, mission goals, and software architectures from which applications able to interface SmartSoft and ROS [25] can be generated.

Other research in robotics focuses on modelling robotic missions. For instance, by proposing better comprehensible mission DSLs that facilitate their definition by non-programmers [10,53] or through identification of specification patterns to further ease robot mission planning [54,55]. The application of block-based modelling languages also is subject of is addressed by ongoing research in robotics too [56–58]. While being less domain-specific than mission planning languages, this research trend holds promise for easing robot programming in the future.

Overall, there is a plethora of publications on the application of model-based and model-driven techniques to robotics, there only are few studies on this.

Summarizing, the use of MDE in robotics looks promising as testified by many initiatives, works, and statements made by organizations. However, there is no reproducible evidence on which types of robots are supported by existing MDE approaches, which type of support models provide,

how existing MDE approaches are validated, and how tools support existing MDE approaches. To better understand all these aspects, we performed this mapping study that will be detailed in the remainder of the paper.

3 Study design

We designed and performed this study following the established guidelines for systematic mapping studies [18,19]. Based on these, this section introduces the core aspects of the design of our mapping study. Specifically, we present the research goal and questions driving the work and describe the various steps of our research method.

A complete *replication package* is publicly available⁴ for independent replication and verification of our study. The replication package includes the raw data of our search and selection phase, the list of selected primary studies, the raw data extracted from each primary study, and the R scripts we developed for data exploration and analysis.

3.1 Research goal and questions

The goal of this study is to identify, classify, and evaluate trends, focus, and open challenges in existing research in MDE for MRSs. We are interested in the points of view of both researchers and practitioners.

With this goal in mind, we applied the population intervention comparison outcome context (PICOC) criterion in accordance with [59], and a set of research questions that address different aspects of the state-of-the-art of MDE for MRSs. Our PICOC criterion definition is the following:

- POPULATION: Mobile Robot Systems
- INTERVENTION: Model-Driven Engineering approaches for Mobile Robot Systems
- COMPARISON: not applicable
- OUTCOMES: a classification of the primary studies that reflects the current state-of-the-art of MDE for MRSs
- CONTEXT: academic peer-reviewed publications with a software engineering perspective

The research questions to be answered for achieving our goal are:

RQ1: What are the **publication trends** of research in MDE for MRSs? Rationale: Research in academia is a multi-faceted and dynamic ecosystem, where several research groups investigate specific scientific problems over time with different levels of independence

⁴ <https://github.com/S2-group/MDE4MRS-Replication-package>.

and different methodologies, as well as targeting different forums and venues for dissemination and communication. Outcome: A snapshot of (1) the set of academic venues and forums where research ideas and results concerning MDE for MRSs, and (2) the density of scientific interest in MDE for MRSs.

RQ2: Which **types of MRS** are supported by existing MDE approaches? Rationale: There are several types of MRS that have been engineered using MDE approaches. Outcome: The set of characteristics of MRSs engineered using MDE approaches.

RQ3: How do existing MDE approaches **support the engineering of MRSs**? Rationale: Different MDE approaches for MRSs focus on different engineering aspects as well as different characteristics of MRSs. Outcomes: (1) A mapping of how MDE approaches support the engineering of MRSs, with special attention to: entailed modelling languages and automation mechanisms (i.e. model transformations), and (2) the set of aspects of MRSs that are engineered by MDE approaches (e.g. behaviour, movement, image recognition, pathfinding, architecture).

RQ4: How are researchers **validating** existing MDE approaches for MRSs? Rationale: MDE approaches for MRSs can have different characteristics and support different types of MRSs. Outcome: A precise characterization of (1) the types of validation applied to the approaches and (2) the types of MRSs used as subjects during the validation of the approaches.

RQ4.1: Which **types of validation** have been applied to existing MDE approaches for MRSs? Rationale: Research results can undergo different types of validation and evaluation, as described by Wieringa et al [60]. Outcome: A map that illustrates how MDE approaches for MRSs have been validated or evaluated (e.g. industrial case-studies, experiments in a lab, etc.).

RQ4.2: Which **types of MRS** have been used when evaluating existing MDE approaches for MRSs? Rationale: There exist several types of MRS that have been used when evaluating MDE approaches for MRSs. Outcome: The set of characteristics of MRSs used for the evaluation of MDE approaches for MRSs.

RQ5: What does the **landscape of tools** supporting MDE for MRSs look like? Rationale: One of the pillars of MDE is the provision of automated mechanisms for alleviating engineering effort. Automation is provided by specific tools. Outcome: A catalogue of tools (and tool types) supporting MDE approaches for MRSs.

3.2 Search and selection process

Figure 1 illustrates the main phases of the search and selection process of this study, together with the number of potentially relevant studies involved in each phase.

1. Initial search. In this phase, we perform automatic searches on an heterogeneous set of electronic databases and indexing systems [61]. As suggested by Kitchenham et al. [61], to cover as much potentially relevant literature as possible, we choose multiple data sources; in our case they are five of the largest and most complete scientific databases and indexing systems in software engineering: IEEE Xplore Digital Library, ACM Digital Library, SCOPUS, Science Direct, and Web of Science. The search is performed considering title, abstract, and keywords of the considered entries.

(mobile OR driv OR cruiss* OR rover OR ground OR *water* OR aer* OR fly* OR sail*)*

AND

(unmanned OR self OR autonomous OR robot OR vehicle*)*

AND

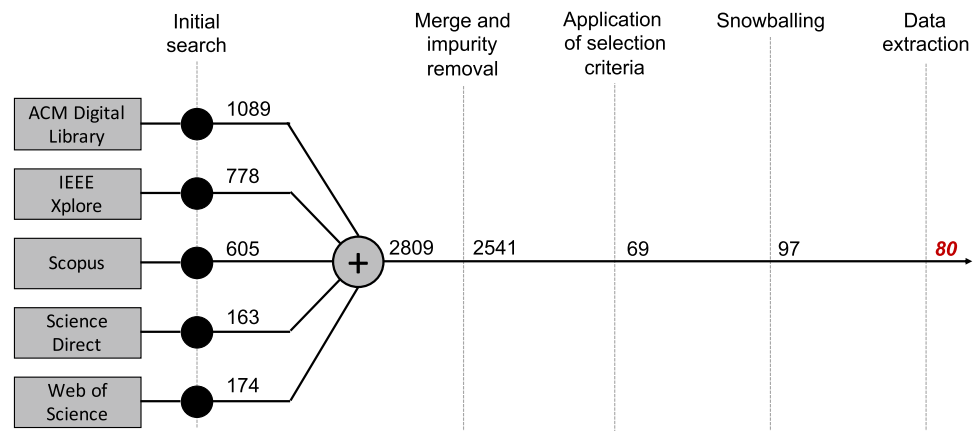
(mde OR mde OR mde OR mdsd OR mbd OR meta-model OR metamodel OR dsl OR domain-specific OR dsml OR model-driven OR model driven OR model-based OR model based)

Listing 1 Search string used for automatic research studies.

The search string used for querying the above-mentioned data sources is shown in Listing 3.2. In order to keep our focus centred around the main goal of the study, the search string has been created by considering our goal and research questions. Specifically, our search string is composed of three main components linked by a logical and relationship, each targeting a specific facet of MRSs: the first component (line 1 in the listing) targets the *mobility* aspect of MRSs, the second component (line 3) targets the robotics domain, and the third component (line 5) targets the model-driven engineering discipline. In order to be reasonably confident about the coverage of our search string, within each of its three components we (1) include keywords related to its corresponding facet, based on the pilot studies and known primary studies, (2) add synonyms to the already added terms, (3) add alternative spellings and acronyms of the already added terms, and finally (4) link all the added terms via a logical or relationship. To ensure consistency, the search strings is always applied to the title, abstract and keywords of papers. Note that the search string was adapted and, in some cases, broken down into multiple strings to comply to the constraints of individual databases or indexing systems (refer to the replication package for more details). This phase produced an initial corpus of 2809 potentially relevant studies.

2. Merging and impurity removal. Due to the nature of the electronic databases and indexing systems, the search results can also include items that were not research papers, such as complete conference and workshop proceedings, international standards, textbooks, editorials, and similar.

Fig. 1 Our search and selection process comprises five phases to search, identify, and process relevant publications



Consequently, in this phase, we manually remove these results as well as duplicates to obtain a coherent corpus of 2541 potentially relevant research studies. Eventually, the remaining papers from the various databases are combined into a single dataset.

3. Application of selection criteria. In this phase, we consider all the potentially relevant studies and filter them according to a set of well-defined selection criteria. The used selection criteria are reported below.

Inclusion criteria

- I1: Studies applying MDE methods or techniques for MRSs. We consider the definition of methodology, given by [62], as a set of methods and techniques, possibly supported by a tool.
- I2: Studies focusing on MDE for MRSs analysed from a software engineering for robotics⁵ perspective. For example, we do not consider studies that cite MDE as a background, but actually focus on other aspects, such as mechanical or electrical engineering.
- I3: Studies including a kind of evaluation or validation of the proposed MDE method or technique. For example, real-world software projects, empirical examples, case study, experiment, formal analysis.
- I4: Peer-reviewed studies: journal papers, workshop papers, conference papers.
- I5: Studies written in English.
- I6: Studies are available in full text.

Exclusion criteria

- E1: Secondary literature studies.
- E2: Tutorial papers, short papers, poster papers, editorials, books, keynotes, tutorial summaries, tool demonstra-

tions and panel discussions, books, introductory papers for books and workshops, technical reports, and any non-peer-reviewed publication.

- E3: Studies not in the scope of software engineering for robotics.
- E4: Studies published before 2000. Since MDE is generally regarded to be born after the OMG standard for the model-driven architecture proposed in 2000 [63].

To be included in the corpus of primary studies, a potentially relevant study must fulfil *all* inclusion criteria and *none* of the exclusion criteria. For each study, an adaptive reading depth was applied in order to carry out the selection process in a time-efficient and objective manner [18]. Specifically, in the first round, the title of the study is considered, enabling us to discard all those studies that clearly do not fall in the domain of MDE for MRSs. In the second and third round, we consider (1) the abstract and keywords and (2) introduction and conclusions of the study, respectively. Finally, if a solid decision about the selection of the study was not taken yet, the full text of the study is inspected.

In order to reduce possible biases, three researchers were involved in this phase of the study. Firstly, we piloted our selection procedure by having three researchers independently assessing a random sample of ten potentially relevant studies; for all ten pilot studies, all researchers agreed (i.e. one included study and nine excluded). Based on the promising results, two researchers proceeded with the independent assessment of all the other potentially relevant studies. Following the suggestion by Wohlin et al. [64], we computed the inter-researcher agreement using the Cohen Kappa statistic and reported it as a quality assessment of this stage of our study. We achieved a value of 0.84 for the Cohen Kappa statistic, which is a good indication of the objectivity in which the selection has been performed which, being ≥ 0.80 , indicates successful objectivity of our selection process according to Wohlin et al. [64]. Finally, all conflicts were resolved by dis-

⁵ According to the definition by the IEEE RAS TC for Software Engineering for Robotics and Automation—<https://www.ieee-ras.org/software-engineering-for-robotics-and-automation>.

curring them with a third researcher, who played the role of arbiter so to avoid endless discussions [65].

The selection procedure described above reduced the corpus to 69 primary studies.

4. Snowballing To mitigate potential bias with respect to the construct validity of the study, we complement the previously described automatic search with a snowballing activity [66]. The main goal of this phase is to enlarge the set of potentially relevant studies by considering each study selected in the previous phases and focusing on those papers that either cited or were cited by it. Note that we perform a closed recursive backward and forward snowballing activity, as shown in [67].

After snowballing, we identify 97 primary studies (and we use them for answering RQ1). However, among those 97 primary studies, some of them were about the same approach, simple extensions of preliminary studies (already included in our work), and similar minor variations. Consequently, we collapsed those 97 studies into 80 unique primary studies

for research questions investigating the content of the studies (e.g. approaches, robot types, applied MDE methods, etc.).

3.3 Data extraction

Through this activity, we create a **data extraction form** to be used to collect data extracted from our primary studies. The form is composed of five facets, each of which addressing a specific research question. Specifically, for answering RQ1 (i.e. research question about publication trends), we consider standard information, such as title, authors, and publication details of each study. As for the other research questions, (i.e. RQ2, RQ3, RQ4, RQ5), we follow a systematic process based on the *keywording* technique [18] for: (1) defining the parameters of each facet of the data extraction form, and (2) extracting data from the primary studies accordingly. The main benefit of the keywording technique is that it allows us to effectively create an extraction form that fits existing studies and tools, and which takes their characteristics into account [18]. The steps of this process are:

1. *Identify keywords and concepts.* We collect keywords and concepts by reading the full text of each pilot study. After all the pilot studies were analysed, the extracted keywords and concepts were combined together to clearly identify the context, nature, and contribution of research on MDE for MRSs. The output of this stage was the full set of keywords and concepts extracted from the pilot studies.
2. *Cluster keywords and define parameters.* We perform a clustering operation on collected keywords and concepts in order to organize them according to the identified characteristics. The clustering operation is very similar to the sorting phase of the grounded theory methodology [68].

Starting from all the keywords and concepts collected in the previous phase, we iteratively group them until saturation of all the concepts is achieved. The output of this stage is the initial data extraction form containing all the identified parameters, each of which represents a specific characteristic of an MDE approach for MRSs.

The following steps are performed individually for each primary study.

3. *Extract data from the current study.* We first extract information about the current primary study to be analysed and then collect information based on the parameters of the data extraction form. In some cases, a primary study can have multiple values for a single parameter. For example, multiple automation techniques can be used within the same approach (e.g. model transformation and automated analysis), multiple types of robots can be used when validating an approach (e.g. aerial and aquatic robots together), and so on. When needed, this aspect is highlighted in the paper. Finally, we collect any kind of additional information that is deemed relevant but that does not fit within the data extraction form.
4. *Refine the data extraction form.* We review the collected additional information because of one of these reasons:

- The collected information is not interpreted correctly. In this case, the collected information is refined.
- The parameters of the data extraction form are not representative enough for the considered primary study. In this case, the data extraction form is refined so that it better fits the collected information. Previously analysed primary studies are reanalysed according to the refined data extraction form.

The above process is complete when all primary studies are analysed. Once we have a complete data extraction form, we isolate the facet related to the technical characteristics of the approaches (i.e. the facet related to RQ3). This is then analysed further to establish any relationships between the parameters, and also to determine the multiplicity and possible values of each parameter. This process is performed through several iterations and leads to the definition of a **classification framework** that can be used by researchers and practitioners for classifying, comparing, and evaluating solutions for MDE of MRSs in an objective manner. The classification framework is summarized in Table 1.

3.4 Data synthesis

We designed and performed the data synthesis activity by following the recommendations by Cruzes et al. [69]. More specifically, our data synthesis activity is divided into two activities: vertical analysis and orthogonal analysis.

Table 1 Classification framework of our study follows its research questions

RQ	Parameter name	Description	Value example
RQ2	Types of supported MRS	The type(s) of robot supported by the approach	Terrestrial MRS, generic MRS
	Cardinality of supported MRSs	The number of robots for which the approach is suited	≤ 5
RQ3	Types of modelling language	The types of language defined in or used by the approach	UML-based language, DS(M)L
	UML profiles	In case of UML-based languages, it identifies the specific language (i.e. UML profile)	SmartSoft, RobotML
	Supported automation	The types of automated mechanisms provided by the approach	Code generation, analysis
RQ4	Engineered aspects	The aspects of MRSs engineered by the approach	Navigation, behaviour
	Types of validation	How the approach was validated in relation to the validation taxonomy by Wieringa et al. [60]	Experiment, case-study
	Types of MRS used for validation	the type(s) of robot supported by the approach	Terrestrial MRS, generic MRS
RQ5	Cardinality of MRSs used for validation	The number of robots with which the approach has been validated	≤ 5
	Language workbench	The workbench related to the modelling languages identified in RQ3	EMF, Xtext
	Eclipse-based tooling	Whether the provided or used tooling is Eclipse-based	Yes, No
	Tools	The names of the tools entailed by the approach	ROSMOD, FLYAQ

In **vertical analysis**, we analyse the extracted data to find trends and collect information about each parameter of our classification framework. In this case, we apply the *line of argument* synthesis [64]; that is, we first analyse each primary study individually in order to document it and to tabulate its main features with respect to each specific parameter of the data extraction form. Next, we analyse the set of studies as a whole, in order to reason about potential patterns and trends. We present the results of this vertical analysis in Sect. 4.

In the **orthogonal analysis**, we analyse the extracted data to explore it and to uncover the existence of possible interesting relations between data pertaining to different aspects of our research. For this purpose, we use contingency tables analysis.⁶ The results of the orthogonal analysis are presented in Sect. 5.

In both cases, we perform a combination of content analysis [70] (mainly for categorizing and coding approaches under broad thematic categories) and narrative synthesis [71] (mainly for detailed explanation and interpretation of the findings coming from the content analysis).

4 Results

In this section, we present the results of this study grouped by research question. For each research question, we report both quantitative data and an interpretation of the obtained results. Primary studies are identified with a unique code, **P_x** (where $1 \leq x \leq 97$, see Appendix A for the complete list).

4.1 Publication trends (RQ1)

For answering RQ1, we consider *all* selected publications after the snowballing procedure, independently of the merging operation we performed on studies about the same MDE approach. So, here we are considering a total of 97 publications, including both the final set of 80 primary studies and the 17 studies which have been merged during the data extraction phase (see Sect. 3.2). We merge those studies to prevent skewing with quantitative results without missing potential unique details carried by these redundant studies.

In addition to standard bibliographic data like authors, title, and year (they are all available in the replication package), for each of the 97 publications, we consider the following information: publication venue, research type, and contribution type.

⁶ Contingency tables may also support dependency analysis between columns/rows/cells done using statistical tests as well as odds ratios. Since our data was overwhelmingly disproportionate, with only few data points per category, it did not make sense to use statistical tests.

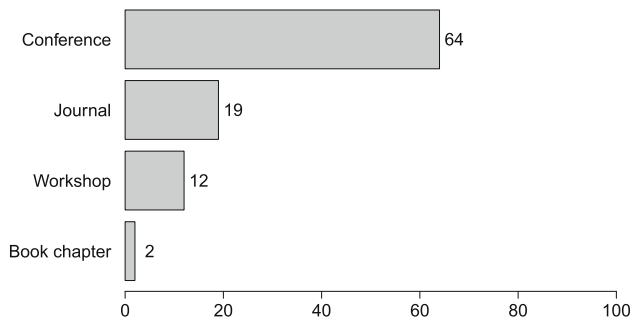


Fig. 2 Publication venues of primary studies

As shown in Fig. 2, the most targeted **publication venues** are conferences (64/97), followed by journals (19/97), workshops (12/97), and book chapters (2/97).

In Table 2, we report the publication venues which have been targeted at least twice by researchers (the last row of the table is an aggregate of all the publication venues with less than two publications). SIMPAR and IROS are by far the most targeted publication venues (11/97 and 6/97, respectively), followed by IRC (3/97); those top-3 targeted venues are specific to the robotics domain. The other targeted venues are either on software engineering (i.e. ASE, EASE, ACM SE), at the intersection of software engineering and robotics (i.e. JOSER and RoSE), or about industrial automation technologies (i.e. INDIN). We can consider the venues mentioned above as good candidates for future publications on MDE approaches for MRSs. Interestingly, the IEEE International Conference on Robotics and Automation (ICRA) does not appear among the most targeted publication venues despite being the most important conference for robotics. We conjecture that this is due to the fact that researchers working on MDE for robotic systems tend to publish in software engineering venues, from our perspective missing a potential chance of impacting the robotics state of the art and practice. Finally, we notice that a large portion of publication venues have been targeted by researchers in MDE for robotics only once. This may be an indication of the scientific interest around MDE techniques across many orthogonal research areas and domains, such as aerospace, multiagent systems, artificial intelligence, communication and networking, self-adaptive systems, systems engineering. Alternatively, this may be due those research works not gaining momentum in these venues or that the results were very specific and hard to build upon. Future analysis of the field could detail this investigation.

Regarding **research types**, we classify each primary study into one or more of the 6 categories reported in Table 3. Those categories have been proposed by Wieringa et al. [60], and we chose them because (1) they are widely used and accepted in various secondary studies in software engineering (e.g. [72–

Table 2 Targeted publication venues

Publication venue	Acronym	#Studies
International conference on simulation, modelling, and programming for autonomous robots	SIMPAR	11
Intelligent robots and systems	IROS	6
International conference on robotic computing	IRC	3
Journal of software engineering in robotics	JOSER	2
International conference on automated software engineering	ASE	2
International workshop on engineering of autonomic and autonomous systems	EASE	2
International conference on industrial informatics	INDIN	2
ACM southeast conference	ACM SE	2
International workshop on robotics software engineering	RoSE	2
International workshop on domain-specific languages for robotics	DSLRob	2
Others	–	69

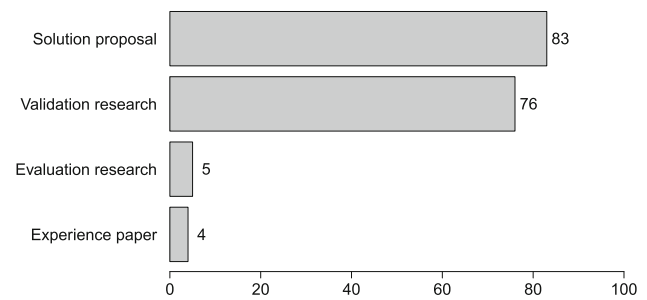
75]), and (2) they can be identified in the primary studies in a cost-effective manner [18].

As shown in Fig. 3, the majority of primary studies involve a solution proposal (83/97), immediately followed by validation research.⁷ The “solution + validation research” is a very recurrent pattern among our primary studies and this unveils one of the major points of improvement for the MDE community working on robotics: proposed approaches are very often not validated in practice (e.g. via an industrial case study or using robots in a production environment). If on one side, we can understand that when a new approach is proposed, the first immediate step is to prove its applicability or feasibility in a controlled environment, on the other we call the community conducting research on MDE for robotics to close the loop from a methodological perspective and apply their proposed approaches in practice. This activity will surely help the overall research field in terms of credibility, industrial adoption and, more importantly, it can

⁷ It is important to note that the sum of all occurrences of research types is more than 97 because a primary study can belong to multiple categories. This applies to the following parameters: types of research, types of contribution, types of modelling languages, engineered aspects, supported automation, and types of MSRs used as validation subjects.

Table 3 Categories of research types (adapted from [18])

Research type	Description
Evaluation research	The study reports on the use of an approach <i>in practice</i> and a rigorous evaluation is conducted (e.g. via a case study)
Experience papers	The study explains what and how some activity has been done in practice. It has to be the personal experience of the author
Opinion papers	The study expresses the personal opinion of the authors about whether a certain technique is good or bad, how certain activities should be done, etc.
Philosophical papers	The study sketches a new way of looking at existing paradigms by structuring the field in form of a taxonomy or conceptual framework
Solution proposal	The study presents a solution for a problem; the solution can be either novel or a significant extension of an existing one
Validation research	The study investigates the properties of an approach that has not yet been applied in practice. Used techniques include lab experiments, simulations, mathematical proofs

**Fig. 3** Research types of primary studies. Among our primary studies, there are no occurrences of philosophical papers and opinion papers

potentially trigger a fruitful wave of collaborations between academics and industrial practitioners.

The five studies performing evaluation research and the four experience papers provide evidence that some steps are already being done towards the industrial application of MDE approaches for MRSs. As an example of successful evaluation research, P19 presents a modelling infrastructure based on a set of declarative DSLs for specifying and executing robotics tasks. Models conforming to those DSLs are transformed into a component-based implementation integrated with a third-party planner for suitably executing the modelled tasks. The whole approach has been applied for a week in the Katharinen Hospital in Stuttgart, where the realized system faced daily hospital routines and operated between real patients and caregivers.

We characterize the **contribution types** of each publication by reusing the categorization proposed in Petersen et al. [18]. According to such categorization, each primary study can provide one or more of the contribution types presented in Table 4.

As shown in Fig. 4, the main contribution types are *method* (97 occurrences), *tool* (71 occurrences), and *model* (58 occurrences). This result does not come as a surprise since the MDE paradigm is fundamentally rooted in the (regulated and/or formalized) use and manipulation of *models*, predominantly via automated tool support (e.g. for models transformation, analysis, and interpretation).

Open items are discussed in 4 primary studies (P74, P75, P84, and P89). Those studies have been published between 2015 and 2017 and the described open items are mostly about how the proposed method and technique can be extended and/or enhanced. The lack of papers exclusively dealing with the elaboration of challenges in MDE for mobile robotic systems may be seen as an indication of the actual need of scoping studies like this one, which may help researchers in objectively identifying open research gaps in the field.

Four primary studies are contributing specific *metrics* for engineering MRSs via MDE techniques (P1, P2, P43, P53). For example, the authors of P43 present an approach supporting the integration of safety techniques within an MDE-based

Table 4 Categories of contribution types (adapted from [18])

Research contribution	Description
Method	Presents general concepts and working procedures to address specific concerns about MRSs
Metric	Presents specific indexes and measures to assess certain properties of a MRS
Model	Presents information, representations, and abstractions to be used when engineering a MRS
Open items	Presents the relevant research challenges to be addressed when dealing with MRSs
Tool	Presents any kind of developed tool or prototype for engineering MRSs

process for robotic systems. The proposed approach is realized as a methodology and an Eclipse-based framework in which safety engineers can use the concepts and metrics of the IEC61508 standard for functional safety [76]. The framework is able to automatically assess the safety of robotic systems [1] in their early development phases. By quoting the authors of P43, *the use of the proposed methodology [fills] the gap between system modelling and safety assessment tools and helps to better cope with system engineering time and cost constraints*. Even though we found only four primary studies proposing metrics for MRSs, this can be seen as a positive trait of the MDE community since it shows the interest in the measurement and assessment of MRSs from some perspectives of interest (e.g. safety by design).

4.2 Types of supported MRSs (RQ2)

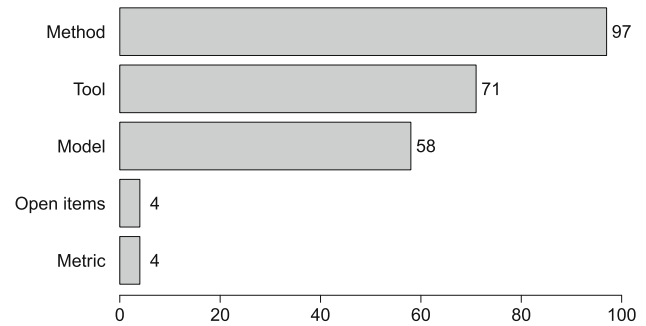
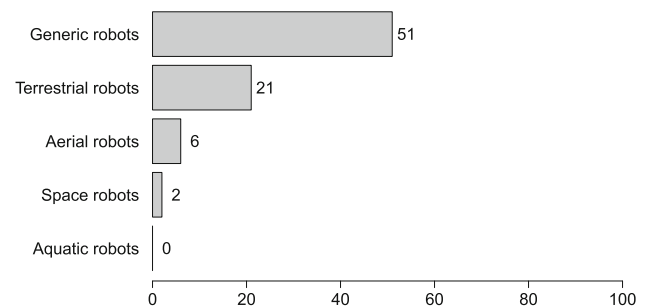
This research question focuses on the types of robots supported by the analysed MDE approaches for MRSs and on whether the engineered approaches involve either multiple robots or a single robot.

For extracting the **types of considered robots**, we considered only studies where the authors explicitly claimed that their proposed approach is specific to robots with certain characteristics (e.g. unmanned autonomous vehicles—P40) and clustered such studies according to the mentioned type of robot (e.g. aerial robots).

Based on our data extraction strategy, we identified five categories of robots, namely terrestrial robots, aerial robots, aquatic robots, space robots, and generic⁸ robots that are addressed in the corpus. In addition, we keep track of those primary studies where the proposed MDE approach is generic, i.e. it is not tailored to any specific type of robot and thus in principle, it is applicable to any robotic system.

Figure 5 shows that the vast majority of contributed approaches is related to *generic* robots (51/80 primary studies). This result can be seen as an indication of the high level of flexibility of MDE methods and techniques, which can be applied to different types of MRSs, while their inner mechanisms remain at a high level of abstraction. The flexibility

⁸ We categorize mobile robots as *generic* when a specific category (e.g. aerial) is not explicitly mentioned nor extractable from the study.

**Fig. 4** Contribution types of primary studies**Fig. 5** Types of robots

of generic MDE approaches is also confirmed in practice by the fact that among them, we can find two of the five studies performing evaluation research and one of the four experience papers (see Fig. 3). This is followed by terrestrial robots (21/80 primary studies), which includes three studies on humanoid robots.

In some cases, generic approaches trade the independence from the types of robots with the types of supported missions. For example, the modelling platform described in P11 achieves genericity by means of three domain-specific modelling languages: MML (Monitoring Mission Language), RL (Robot Language), and BL (Mission Behaviour Language). MML allows non-technical operators to specify missions as sequences of tasks, task dependencies, forks/joins, areas, and obstacles; RL allows robot engineers to extend the generic platform and specify the technical characteristics of new types of robots (e.g. hardware configuration, sensors); BL permits to specify the chain of atomic movements of each

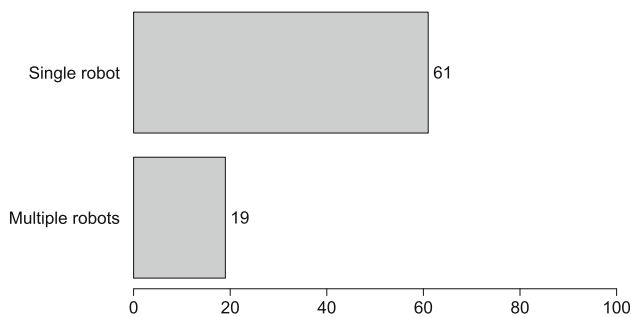


Fig. 6 Cardinality of robots

individual robot declared in MML and RL models. The languages above have been designed so to allow the platform to automatically generate BL models from MML and RL models, thus (1) relieving the operator from the intricacy of low-level details of the individual robots and (2) making the whole platform independent of the types of used robots. Here, the trade-off lies in the fact that P11 supports the definition of only civilian monitoring missions, e.g. monitoring of forestry, control of pollution emissions, surveillance of public events.

Among the approaches supporting specific types of robots, *terrestrial robots* are the most supported ones (16/80 primary studies), followed by other types of approaches with much lower occurrences. For example, P44 proposes a domain-specific modelling language for formally specifying constraint-based programming tasks. The DSL is specific for service robots and provides constructs such as *robotic scene*, *location*, *object*, etc. Models conforming to such a DSL can be used as starting points for automatic model verification and code generation targeting different platforms (both hardware and software). The DSL comes with a textual concrete syntax and promotes reuse and component-based development of robotic systems by grouping potentially reusable functionalities.

No approaches were specific to aquatic robotic systems (e.g. underwater robots). Nevertheless, aquatic robots have been used twice as subjects during the validation of generic MDE approaches (see Sect. 4.4.2). This is a further confirmation of the flexibility and applicability of MDE methods and techniques in the robotics field.

When dealing with the cardinality of robots, we classify each primary study according to whether its proposed approach assumes that the system is composed of multiple robots or a single robot.

As shown in Fig. 6, More than two thirds of the approaches support *single robots* only (61/80 primary studies) but there is also a non-negligible number of studies supporting systems with *multiple robots* (19/80 primary studies).

This result is promising since supporting more than one robot introduces additional challenges, such as robots coordi-

nation and synchronization, communication issues (e.g. path loss), communication overhead, etc. We speculate that this result may be an indication that MDE techniques can support engineers in abstracting from and managing the intrinsic variability emerging when managing multiple different robots. For example, P16 proposes an OWL-based ontology of mobile robotics capabilities, a UML-based DSL for describing robotic systems with variation points which can be filled by third-party entities, and a code generator targeting several robotic middlewares (e.g. OROCOS [9]) and simulators (e.g. Blender MORSE [77]).

Moreover, MDE techniques can also help to engineer in enabling better reasoning on the high-level problems emerging when dealing with multiple robots, such as the formation of robot ensembles at run-time. For example, P7 proposes a framework for programming distributed mobile robots in such a way that collisions are prevented, and each individual robot moves in the environment according to optimal motion plans. The approach presented in P7 is based on a provably correct multi-robot motion planner which is decentralized, asynchronous, and reactive to dynamically generated task requests. At the core of the motion planner lies P, a state-machine-based DSL for defining event-driven asynchronous programs for the robots. P specifications are verified at design-time using the Zing model checker [78]. Finally, C code is automatically generated from P specifications and directly deployed as ROS [25] nodes in the system.

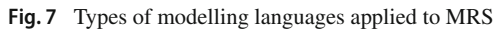
4.3 Applied MDE methods and techniques (RQ3)

4.3.1 Types of modelling language

Observing the types of modelling language used by the analysed approaches, it is not surprising that DSLs, both non-UML and UML-based, own the stage. In some cases, different types of languages are used in combination in order to enable specific kinds of analysis or modelling different aspects of the MRS. Interestingly, some studies report on experiences with different languages for solving a specific problem. An example is P13, where UML Real-time, derived from the ROOM language [79], is compared to AADL [80] for modelling and timing analysis of complex (swarming) problems. In this specific study, the authors suggest that AADL and UML real-time were both suitable for their purpose, so the choice on which one to prefer highly depends on the tools implementing them (Fig. 7).

4.3.2 UML-based languages

Although representing almost half of the languages mentioned in the analysed approaches, UML-based languages are rarely explicitly enclosed in a so-called UML profile. The most often used UML profiles are SysML [81] and Smart-



UML profile	Papers
SmartSoft	P10, P16, P62, P63, P74
SysML	P22, P24, P25, P28
RobotML	P43, P54
UML Real-time	P13
MechatronicUML	P42
MARTE	P68

4.3.3 Engineered aspects

[illegible]

uct lines (three occurrences), etc. This result is expected since several research groups are currently active in applying MDE techniques on MRSs and are targeting publication venues belonging to various computing domains (see Sect. 4.1). We can also notice primary studies investigating on less-explored research venues, such as methods for managing the uncertainty of units of measure at runtime (P1), monitoring functional requirements at runtime (P20), the composition of languages and code generators (P23) and methods for systematically testing autonomous robots (P67). We invite the reader to use those studies as starting points for understanding what MDE techniques are capable of, in addition to more canonical applications like code generation, mission specification, abstraction from the robotic platform, etc. For a complete overview, the interested reader can refer to our replication package, where the engineered aspect of each primary study is available.

4.3.4 Supported automation

 Springer

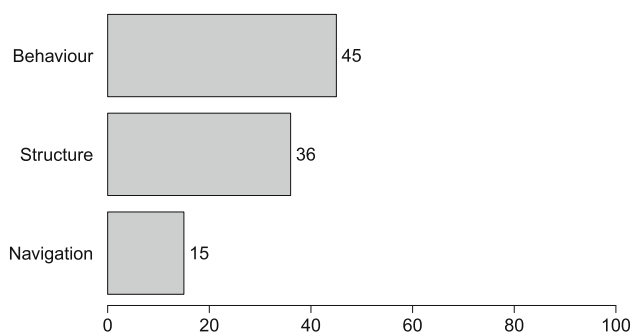


Fig. 9 Cross-cutting aspects of the primary studies

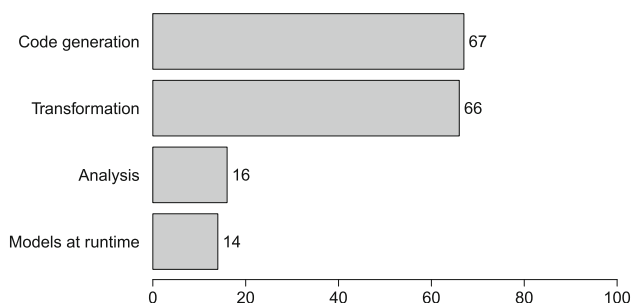


Fig. 10 Automated MDE mechanisms for MRS

mation, code generation, model-based analysis, and models@runtime.⁹

Model transformation and code generation are, as we expected, the most common automation mechanisms carried out from models; only in 3 cases, neither of the two is provided. In 11 cases, it is not specified whether code generation is achieved via transformations, while in 25 cases transformations are employed for other purposes than code generation (14 for model-based analysis and 9 for models@runtime). Only 1 approach (P20) exploits models@runtime in combination with model-based analysis.

4.4 Validation of MDE approaches for MRSs (RQ4)

The objective of RQ4 is to provide a map about (1) how MDE approaches for MMRSs have been validated (RQ4.1—Sect. 4.4.1) and (ii) the types of robots employed during the validation of each approach (RQ4.2—Sect. 4.4.2).

4.4.1 Types of validation (RQ4.1)

The intuition behind RQ4.1 is that approaches that have been (rigorously) validated in industrial/realistic settings are more likely to be adopted in the industry [84]. For example, a val-

⁹ It is important to note that model transformation can be considered a super-type since it is often (but not always!) the way to achieve the other three. In those cases, we marked the primary study as using both model transformation and the other automation technique.

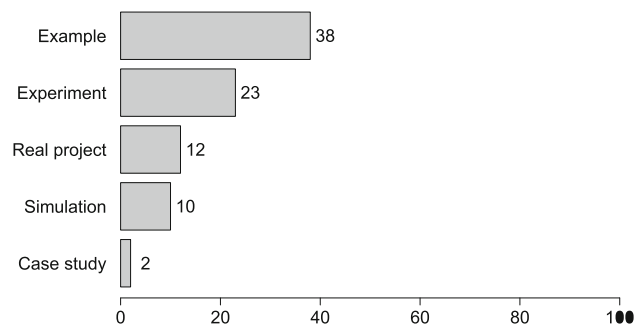


Fig. 11 Types of performed validation

idation carried out only by showing an example of applying proposed approach (e.g. P6) has a much lower degree of realism with respect to a validation performed using real robots deployed in the field (e.g. P76).

As shown in Fig. 11, a large portion of primary studies relies on the least rigours and non-systematic type of validation, i.e. just showing an *example* of the application of the proposed approach (38/80). This result might be exploited by our community in order to raise discussion on how to carrying out research. On one side, it is important for consolidated research results to provide rigorous, replicable, and realistic evidence about the performance of the proposed approach. This would probably increase the likeness to have impact either in academia or industry. On the other side, it is important to consider that early concept papers might not be able to provide a mature validation. It is important to do not rule out these types of works, since they are an important part of the exchange of ideas in science. For what concerns consolidated research results, it would be probably important to strive towards performing well-designed empirical experiments on their proposed approaches, preferably in realistic scenarios and with real robots. This result indicates the need of providing more rigorous, replicable, and realistic evidence for consolidated research results to have a stronger impact both in academia and industry. Nonetheless, it is important to still consider the usefulness of concept and visionary papers, for which providing a mature validation is not feasible. These works are in fact an important part means for exchange of ideas in science, also for practitioners.

Some research groups are already moving towards this direction, with 23 primary studies reporting about empirical *experiments* carried out on the MRS (23/80), with the deployment of the MRS in the context of a *real project* (12/80) or even by applying the *case study* research methodology (2/80) [85]. As an example of realistic validation, the authors of P5 report about their experience in applying their proposed approach on a youBot mobile manipulation robot participating in the RoboCup@Work competition [86].

Also, ten primary studies report about a simulation-based validation of their proposed approach. In those cases, the fol-

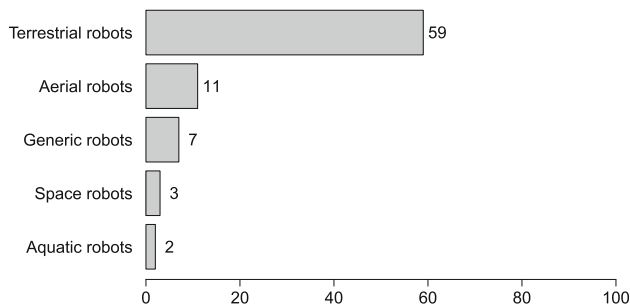


Fig. 12 Types of robots used as subject for validation

lowing simulators are used: Gazebo¹⁰ (P7, P34), MATLAB¹¹ (P13, P59, P71), SimSpark [87] (P30), Player/Stage [88] (P39), JSBSim [89] (P11), and custom simulators (P15, P73).

4.4.2 Types of MSRs used as validation subjects (RQ4.2)

While RQ2 focuses on the characteristics of the robots supported by the approach itself (e.g. the approach presented in P59 works only for swarms of aerial robots), here we focus on the type of robots actually employed while *validating* the proposed approach.

Terrestrial robots (59/80) are the most often used **types of robots** when validating MDE approaches for MRSs (see Fig. 12). This result is expected since terrestrial robots are the ones with the most controllable dynamics. Also, the usage of terrestrial robots is facilitated by the fact that many types of research and educational terrestrial robots exist in the market, such as the Pioneer 3-DX platform¹² (e.g. used in P26), the iRobot Create¹³ (e.g. used in P2), Lego Mindstorms¹⁴ (e.g. used in P78), and similar robots. Nonetheless, in many primary studies, a full robotic prototype has been designed and realized from scratch (i.e. from hardware design up to the application level), such as in P41 (a service robot), P15 (a rover), and P33 (self-driving car).

Aerial robots have been used in 11 primary studies, both in single-robot missions (e.g. P14) and swarm-oriented ones (e.g. P59). Differently from terrestrial robots, among those 11 studies, we can mostly find custom-built vehicles, like in P24 (fixed-wing aircraft) and P40 (quadrotor with Arduino-based controller). The effort for designing, developing, and properly testing autonomous flying vehicles is significant. Therefore, it would be desirable to have a *shared hardware-software platform for aerial robots*, instead of having different research groups building their own ones. This

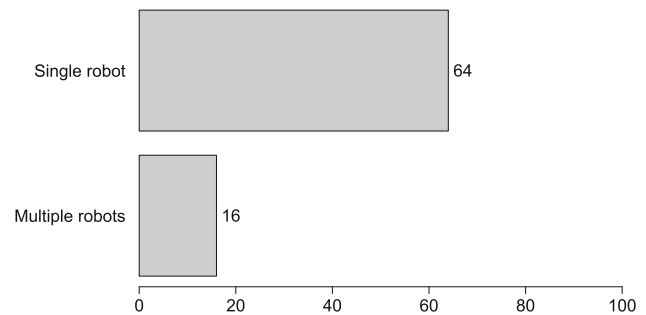


Fig. 13 Cardinality of robots used as subject for validation

would open many opportunities for scaling up our research, including:

- reusing already built and well-maintained flying vehicles,
- having a common (possibly benchmarked) hardware platform across different research groups so to easily compare independently-achieved research results,
- reusing different (open-source) software and hardware modules across research projects (e.g. the PX4 autopilot firmware),¹⁵ and
- collaborating with companies already using the same platform for their commercial products, thus potentially leading to a smoother technological transfer of the obtained research results.

In 7 cases, validation has been performed on *generic mobile robots*. In those cases, the validation of the approach has been performed on synthetically built examples focusing on robotics in general, without specifying the specific type of robot and with a focus on specific aspects of the system (e.g. navigation, planning). If on the one side, we understand that for some studies having a validation on a generic mobile robot is practical (e.g. in case of very preliminary results), then on the other side, we invite researchers to spend effort in the validation of their approaches towards using realistic robotic systems. More realism helps in terms of adoptability of the proposed approach and provides more evidence about the characteristics of the approach, especially when dealing with run-time properties such as performance, energy consumption, and safety.

About the **cardinality of the robots** used as subjects for validation (Fig. 13), we notice a strong imbalance in favour of *single robots* (64/80) at the expense of *multiple robots* (16/80). This result is partially due to the fact that the majority of primary studies are dedicated to single-robot systems (see Sect. 4.2). However, there are 12 primary studies whose proposed approach is applicable to multi-robot systems, but their validation is focusing on a single robot (e.g. P1, P12,

¹⁰ Gazebo: <http://gazebo-sim.org/>.

¹¹ MATLAB: <https://www.mathworks.com/products/matlab.html>.

¹² Pioneer 3-DX: <https://www.generationrobots.com/en/402395-robot-mobile-pioneer-3-dx.html>.

¹³ iRobot Create: <https://www.irobot.com/about-irobot/stem/create-2>.

¹⁴ Lego Mindstorms: <https://www.lego.com/en-us/mindstorms>.

¹⁵ PX4 autopilot firmware: <https://github.com/PX4/Firmware>.

Table 6 Language workbenches used to build DSLs for MRS

Language workbench	Papers
EMF	P8, P11, P27, P28, P35, P43, P51, P52, P53, P56
Xtext	P29, P33, P37, P48
MontiCore	P6, P19, P21, P23
Internal	P5, P44
AToM	P39, P47
MPS	P4

P20). After a closer inspection of all those studies, all of them focus on a specific aspect of a MRS (e.g. how software components communicate—P12), independently of the number of robots involved in the mission (e.g. in P12 software components can be deployed flexibly on both single and multiple robots).

4.5 Tool support (RQ5)

There is a plethora of tools used for various aspects of MDE of MRSs. Interestingly, some of the approaches focus on the language workbench used to develop a specific DSL for MDE of MRSs. Among them, the eclipse modelling framework [90] (EMF), for defining metamodel-based DSLs, is the most popular workbench, followed by Xtext [91] and MontiCore [92], both for textual DSLs. Besides language workbenches (see Table 6), the analysed approaches rely on tools of all kinds, from model transformation suites (e.g. Aceleo)¹⁶ and execution semantics for metamodels (e.g. Kermeta3)¹⁷ to specialized IDEs (e.g. RoboStudio [93]), web-based mission planners (e.g. FLYAQ [10]) and analysis tools (e.g. RTEdge [94]). Among the mentioned supporting tools (74 in total), 29 are based on the eclipse ecosystem.

5 Orthogonal analysis

The orthogonal analysis investigates possible correlations between related concepts of the classification framework. To this end, we cross-tabulated and grouped the data related to these concepts, compared the data relating to the different pairs of concepts, and discussed findings of interest. This section describes these findings. Moreover, we investigated correlations between other pairs of concepts not discussed below. We found these to confirm the trends of the vertical analysis. However, all observations are available from the replication package, which allows further investigation of the data.

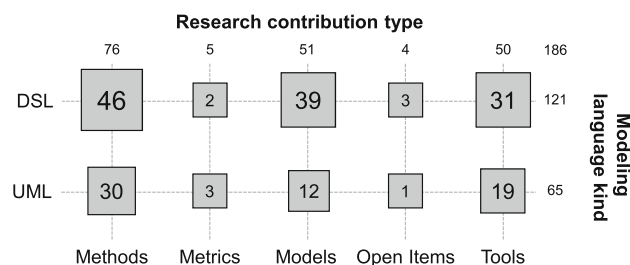


Fig. 14 Modelling language kinds relative to research contribution types

5.1 Modelling languages and research contribution type

In our corpus, DSLs, variants of UML, and other, non-descript, modelling techniques are used as objects in different types of research contributions, including novel methods, metrics, models, open items, and tools. By relating both facets, we aim to understand whether DSLs or UML-based modelling languages are specific to a certain research type. To this end, we compare both dimensions as depicted in 14. As some publications employ multiple modelling techniques for one or more contribution types, the sum of numbers in a column or in a row can exceed the number of publications of our corpus. For instance, the research presented in study P93 contributes novel abstractions based on UML, a method for using these, and a metric to evaluate their use. Consequently, it adds three individual entries to our cross-tabulation as presented in 14. Overall, the research addressed by publications included in our corpus contributes 186 of such entries.

Overall DSLs make up 121 of these entries, whereas UML variants only make up 65. Hence, DSL-based publications contribute to 2.6 research types on average and publications reporting research with UML variants contribute to 2.2 research types on average. DSLs are used to contribute methods and models most often. In contrast, UML-based languages are used to contribute methods and tools most often (Fig. 14).

From our findings, the focus on DSLs for model type research might be due to the heterogeneity of the field which demands more different modelling techniques than the purely functional and discrete UML diagrams. And while specifying

¹⁶ Eclipse Aceleo: <https://www.eclipse.org/aceleo/>.

¹⁷ Kermeta3: <http://diverse-project.github.io/k3/>.

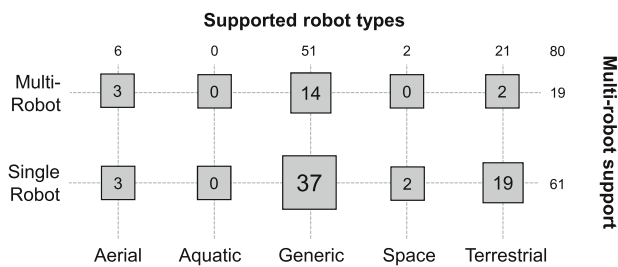


Fig. 15 Multi-robot support relative to supported robot types

ing such techniques generally is possible via UML profiles, the appearance of powerful language workbenches might have raised the number of contributions providing stand-alone domain-specific languages for specific challenges in mobile robotics. In the future, we expect to see more publications contributing specific modelling languages to mobile robotics, which would be in line with a general trend towards DSLs in robotics [95].

This might entail a diversification of robotics DSLs into DSLs for specific kinds of systems in the context of specific application scenarios (e.g. DSLs for search-and-rescue missions with aerial robots or DSLs for modelling logistics operations for terrestrial robots). We expect that most of these will be realized as external DSLs built on top of powerful language workbenches (cf. Table 6).

5.2 Multi-robot support and supported robot types

As presented in Fig. 5, the majority of research on MDE for MRS aims to be generic with respect to the supported robot types (51 out of 80 publications). This also holds for major European research projects in this discipline, which, e.g. aim for establishing best practices for modelling different robotics concerns in a generic fashion¹⁸ or for enabling modular model-driven development in robotics in general.¹⁹

When considering approaches specific to distinct types of robots, we found that research most often focuses on single generic mobile robots (reported 37x), single terrestrial robots (19x), and generic multi-robot systems (14x). Relatively, the small number of approaches supporting terrestrial multi-robot systems is surprising (Fig. 15).

While these findings being are too small to assert significance, the relatively small number of terrestrial multi-robot systems might indicate that the approaches towards MDE for generic MRS demand further evaluation with specific robot types or that the generic multi-robot approaches are good enough to subsume multi-robot research for specific robot kinds.

¹⁸ BRICS FP7: www.best-of-robotics.org.

¹⁹ RobMoSys Horizon 2020: <https://robmosys.eu/>.

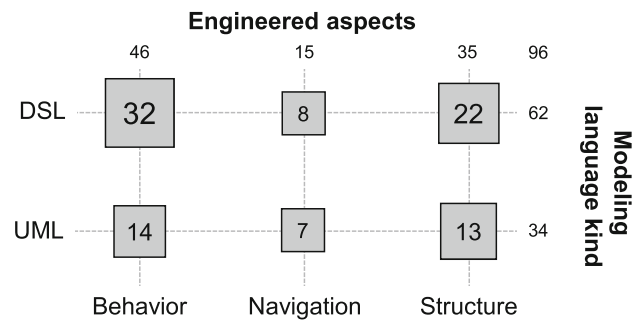


Fig. 16 Engineered aspect relative to employed modelling language kinds

5.3 Modelling language kinds and engineered aspects

To better understand how modelling languages are applied to the different engineered aspects, we cross-tabulated these relative to each other as well. As some publications addressed multiple aspects with multiple kinds of modelling languages, the overall number of 86 findings is slightly higher than the number of publications in our corpus.

Our observations show that DSLs and UML are used differently to address the engineered aspects of MRS. Where aspects were mentioned, they were addressed by DSLs more often (reported 62timesx) than by UML (34x).

Contributions leveraging DSLs focus more often on structure and behaviour (together 54/62), whereas navigation is significantly less often addressed (7/62) by DSLs. In contrast, contributions leveraging UML relatively more often focus on navigation (7/34) (Figs. 16, 17).

Our data cannot explain why this is the case. It could be that navigation is more of a cross-cutting concern that addresses both structure and behaviour, for which the combination of respective UML diagrams might be better suitable than creating a novel DSL that captures both from scratch.

With the large majority (64%) of DSL-related contributions being developed starting 2014, we expect more contributions on DSLs for MRS navigation in the future.

5.4 Engineered aspects and automation support

To uncover how the supported engineered aspects related to the provided automation support, we also cross-tabulated these relative to each other. Again, publications can produce multiple entries to this cross-tabulation, e.g. where code generation is applied to behaviour and structure, two entries are counted.

We found that the engineering aspects of MRS also relate differently to the supported automation techniques. Overall, the publications our corpus report on leveraging various automation techniques 184timesx to address behaviour,

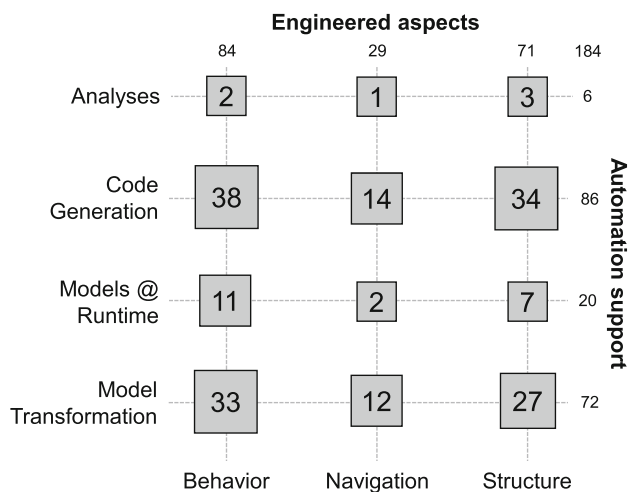


Fig. 17 Engineered aspects relative to the provided automation support

navigation, or structure. According to these reports, code generation (reported 86timesx) and model transformation (72x) are supported most often, whereas models at runtime (20x) and model analyses (6x) are supported significantly less often.

As implied by the use of DSLs and UML to address these engineering aspects, also the automation support for behaviour (84x) and structure (71x) is more prevalent than for navigation (29x). Where behavioural, navigational, and structural modelling contributions are similarly often supported by code generation and model transformation, navigation rarely is relatively more supported by models at runtime.

The lack of models at runtime support for navigation is surprising as navigation demands for some kind of modelling (robot, environment) at runtime. This might indicate that much of navigation still is implemented with traditional programming languages. Overall, MDE for MRS strongly focuses on producing solutions via code generation of model transformation.

We assume that this might be due to the large amounts of infrastructure code that even a simple robotics application needs.

The increasing popularity of a few robotics software platforms (such as ROS [25] or SmartSoft (P10)) that take care of infrastructure challenges, might liberate software engineering research capabilities to focus on the added value of analyses of MRS structure, behaviour, or other aspects.

6 Discussion

We conducted this systematic literature study to gain insights on MDE for mobile robotic systems. To this end, we formulated five research questions (see Sect. 3). This section reports our observations on the main findings,

which are described in Sect. 4, and orthogonal analysis, which is described in Sect. 5.

Regarding the **publication trends** (*RQ1*), we found that most contributions to MDE for MRS are published at conferences, followed by journals, and workshops. This is probably in line with any software engineering publication. Most publications occur at specialist venues dedicated to the intersection of robotics engineering and software engineering, such as SIMPAR, IROS, or IRC (cf. Table 2). This focus is confirmed by the little number of publications appeared in venues purely dedicated to robotics (e.g. the International Conference on Robotics and Automation (ICRA)), model-driven software engineering (e.g. International Conference on Model-Driven Engineering Languages and Systems (MODELS)), or software engineering in general (e.g. International Conference on Software Engineering, (ICSE)). However, we can observe an increasing interest in the software engineering community, and we expect that in the near future in the program of software engineering conferences like ICSE there will be dedicated sessions to robotics.

In fact, many of the challenges identified in the corpus (e.g. modelling of structure, behaviour, navigation) are general enough to warrant publication and presentation at less focused venues, including conferences such as ICRA, MODELS, ICSE, ESEC/FSE, and journals such as Journal on Software and Systems Modelling (SoSyM), IEEE Transactions on Software Engineering (TSE), ACM Transactions on Software Engineering and Methodologies (ToSEM), Automated Software Engineering (ASE), Journal of Systems and Software (JSS), International Journal of Robotics Research (IJRR), IEEE Transactions on Robotics (TOR), or Journal of Field Robotics (JFR). The organization of workshops, such as the workshop on Robotics Software Engineering (RoSE) or the workshop on Model-Driven Robotics Software Engineering (MORSE), at the non-specialist top conferences (ICSE, ICRA, ASE, ESEC/FSE) or the organization of special issues in the related journals are also contributing to maximize outreach, dissemination, and exposure of the community's research.

Research in MDE for MRS largely focuses on solution-seeking research [96], contributing methods and tools in the form of solution and validation research. This is a common trend shown in many mapping studies in the field [1,46,97–99]. However, the relatively large number of publications contributing tools (cf. Table 4), appears to be remarkable.

We can probably find a reason for that in the nature of research on MDE for MRS. In fact, this type of research focuses on answering research questions centred around robotic systems and in producing results that are relevant to the community. Therefore, there is a need for tools that automate the proposed methodologies and make it usable

in practice. The next step will be on making these tools actually usable in everyday life and transferable into industrial practice, for instance providing guarantees for safety aspects [1].

On the supported **types of considered robots** (RQ2), we found a strong focus on research about single generic terrestrial robots. This is not surprising, since single generic terrestrial robots require less maintenance and operation effort with respect to other types of robots. For many research challenges in MDE for MRS, e.g. cognitive robot behaviour, modelling and generating robotic software architectures, implementing robust navigation, multi-robot systems, as well as aerial systems or aquatic systems, introduce additional challenges that might, if not in the focus, prevent advancing the original research challenges.

Despite focusing on single robot systems, a significant number of the studies of our corpus also addresses MDE for multi-robot systems. This might be due to the improved accessibility of individual platforms (e.g. usable drones have become fairly inexpensive) since this enables novel methods to evaluate systems of collaborative or competitive robotics in a wide range of scenarios, such as e.g. in smart factories or for disaster relief.

Moreover, the majority of contributions focuses on MDE for generic mobile robot types (e.g. modelling of robot capabilities (P16)), which generally should support transferring these results to other related domains, such as smart manufacturing systems (which also need elaborate behaviours, structures, as well as object recognition and reasoning capabilities). We believe that this transfer can yield valuable insights into the concurrent challenges specific to MDE for MRS and could motivate future research in this direction.

We want to highlight the need for better validation to clearly identify the limits, scope, and qualities of the proposed solutions [1,100]. It can happen that contributions to single-robot systems might scale to multiple-robot systems, or that generic solutions might actually be less generic than expected.

We believe that this is an important step for having a large adoption in industrial practice.

Concerning the **applied MDE methods and techniques** (RQ3), we found that many publications contribute DSLs, either defined externally or as UML profiles (e.g. RobotML, SmartSoft)—to address challenges regarding behaviour, structure, and navigation. These also include UML profiles not tailored directly to robotics (such as MARTE or MechatronicUML). While this suggests that (tailored) UML generally can be useful to facilitate engineering of MRS, we found that contributions involving

DSLs significantly more often (50 contributions) present new *models* to describe solutions than contributions involving UML (13 contributions),

whereas contributions involving UML more often focus on new *methods* (54 contributions) than contributions involving DSLs (34 contributions).

As new models usually contribute new ways of *conceptualizing things*, whereas methods focus on *doing things*, this might entail that for innovative robotic aspects, DSLs suit better than UML and its profiles.

Also, this would explain why there are significantly more contributions leveraging explicit DSLs than UML. This seems to be a specific characteristic of this domain, which differ, for instance, from for instance, does not

MDE in smart manufacturing [46]. We, therefore, suggest intensifying research on DSLs for specific MRS challenges but advocate keeping the interconnection of all three MRS engineering challenges (behaviour, navigation, structure) in mind when developing such languages to enable their integration with DSLs for interconnected purposes.

Moreover, we observe a strong focus on addressing behavioural and structural challenges instead of navigational challenges. This might be caused by the fact that navigational challenges are relatively uniform across different types of robotic systems, e.g. there need to be sensors and actuators connected to other components realizing the high-level behaviour and most platforms need to take care of local and global navigation issues using laser-based or camera-based orientation.

We also observed a strong focus on constructive automation techniques, i.e. code generation and model transformations over analyses or models at runtime. This is in line with our observation on the constructive nature (solution-seeking) of MRS research (and, hence, the relative abundance of tool contributions). From our experience, one of the major challenges in robotics are configuration and orchestration of the vast numbers of components that need to properly interact to successfully deploy even relatively simple MRS. Not leveraging the main strengths of models, which are in communication and analysis [31] through automated analyses supporting this, hence, is surprising. However, with this increasing wealth of models and modelling techniques becoming available, we suggest that future research also focuses on analyses for, e.g. the reliable or explainable behaviour of robotic systems that support the adoption of MRS outside of the laboratory environments they often are confined to.

Research on MDE for MRS is mostly **validated** (RQ4) by example using single terrestrial robots. This often has the form of non-systematic validation with little rigour, e.g. by giving examples for its application.

We assume this is due to the complexity of performing rigorous experiments that systematically cover (or rule out) all potentially relevant aspects of the system under investigation and its environment. This implies a need of modelling solutions able to grasp and represent various

engineering aspects and stakeholders' concerns, including hardware, mechanics, software, and even the environment, to facilitate simulations and the definition of rigorous experiments.

The choice of terrestrial, single robot, evaluation platforms follows the trends on the kinds of MRS addressed and, hence, is not surprising.

Indeed, research on generic mobile robots should be evaluated with different kinds of MRS. This application of generic solutions to specific robot types might provide insights into limitations not considered when conceiving generic modelling techniques (as often, “the devil is in the details”). In particular, aerial robots have become inexpensive enough to consider as alternatives to terrestrial robots for validation. Indeed, the rigorous experimental evaluation with aquatic and space robots requires to access (realistic) operation environments (e.g. oceans, lakes, space); in this sense, terrestrial robots can be operated more easily in realistic environments.

The main pillars of MDE are abstraction and automation. Consequently, we expect MDE for MRS to provide tools build upon existing **automation tools**

(RQ5). DSLs are quintessential to MDE leveraging automation. This is in line with many studies of our corpus providing novel models or methods using DSLs. Many of these studies leverage sophisticated language engineering tools, such as EMF [90], MontiCore [92], and Xtext [91], which appear to be the most prominent language workbenches to support DSL engineering for MRS.

Moreover, we have found many studies leveraging MDE by using models (e.g. to describe robot structure, behaviour, or navigation) instead of developing novel DSLs. For the use of models, we have found specific IDEs ranging from full-fledged IDEs, such as RobotStudio [93], to web-based mission planners, such as FLYAQ [10], to very specific analysis tools, such as RTEdge [94].

Generally, aside from language workbenches, however, and few DSLs (such as SmartSoft or RobotML) there seems to be little tool reuse in MDE for MRS. Our data does not indicate whether this is due to the heterogeneity of research on MDE for MRS, although the focus on terrestrial single robot systems appears to contradict this assumption, or due to the complexity of the different technological spaces that these more established tools reside in. Moreover, this might be due to established robotics middleware (such as ROS) not being model-driven but instead focusing on traditional programming. Consequently, when reusing such a middleware, there is little “MDE tool” reuse. To uncover why model-driven middleware solutions are used less often, we suggest further research targeting the replication and evaluation of approaches within their, to be explicated, validation frames.

7 Threats to validity

In the following, we discuss the potential threats to the validity of this study as well as the activities we carried out to mitigate them.

7.1 External validity

External validity refers to the generalizability of our results [64].

The generalizability of our results strongly depends on the extent to which the selected primary studies are representative of the research landscape about MDE approaches for mobile robotic systems. This potential threat to validity is mitigated by targeting multiple data sources, i.e. ACM Digital Library, IEEE Xplore, Scopus, Science Direct, and Web of Science. The rationale for having multiple data sources is that their provided results complement each other in terms of coverage of the state of the art. The used data sources cover the area of software engineering well [101,102] and are five of the largest and most complete scientific databases and indexing systems in software engineering. Moreover, we further complement the results of the automatic searches by performing a closed recursive backward and forward snowballing activity.

Moreover, only studies published in the English language have been selected in our selection process (see I5 inclusion criterion in Sect. 3.2). This decision may result in a possible threat to validity because potentially important primary studies published in other languages have not been selected in our research. However, the English language is the most widely used language for scientific papers, so this bias can be reasonably considered as minimal.

In this study, we exclusively target peer-reviewed studies (e.g. journal or conference papers), leaving out the grey literature (e.g. white papers, technical reports, blog posts). This potential threat to validity is intrinsic to our study design because we target high-quality scientific papers, and thus undergoing a rigorous peer-reviewed publication process is an accepted requirement for this kind of scientific works. The same line of reasoning applies for our E2 exclusion criterion, dealing with the exclusion of tutorial papers, short papers, poster papers, editorials, etc., since they do not contain enough information for answering our research questions.

7.2 Internal validity

Internal validity refers to the level of influence that extraneous variables may have on the design of the study [64].

In this case, having a rigorously defined protocol with a rigorous data extraction form helped in mitigating biases related to the internal validity of our research. Also, for what concerns the data analysis validity, the threats are minimal

since we employed only descriptive statistics when dealing with quantitative data. When considering qualitative data, we systematically applied the keywording method for transforming qualitative data into quantitative data. Finally, ten primary studies have been randomly selected and two researchers checked whether the results were consistent, independently from the researcher performing the extraction; moreover, each disagreement has been discussed and resolved, together with a third researcher, when needed.

Moreover, we formalized the design of this study into a detailed research protocol *beforehand* and thoroughly discussed it prior to the actual execution of the study. Our research protocol (and thus the whole study) is based on well-accepted guidelines for systematic literature reviews and mapping studies [19,64,103].

To allow independent replication and verification of our study, a complete replication package is publicly available. The replication package includes the review protocol, the raw data of our search and selection phase, the complete list of selected primary studies, the raw data extracted from each primary study, and the R scripts we developed for data exploration and analysis.

7.3 Construct validity

Construct validity refers to the selection of the primary studies with respect to the research questions we aim to answer [64].

As already discussed in Sect. 7.1, the initial automated search has been performed on six different data sources and complemented with the snowballing procedure. In this context, one potential threat to validity may be caused by a malformed search string. We mitigated this potential threat to validity by (1) following a rigorous process for defining it (see Sect. 3.2) and (2) piloting the search string in preliminary queries using IEEE Xplore.

After having collected all relevant studies from the automatic search, we rigorously screened them in a collaborative fashion and according to well-documented inclusion and exclusion criteria (see Sect. 3.2).

7.4 Conclusion validity

Conclusion validity refers to the relationship between the extracted data and the obtained results [64].

First of all, third-party researchers can verify and check the obtained results by replicating our study independently from our results. This is possible thanks to the detailed research protocol and the complete replication package, which are both available publicly.

Another potential threat to conclusion validity may lie in the definition of the classification framework. Indeed, other researchers may identify classification frameworks with dif-

ferent facets and parameters, thus potentially leading to totally different perspectives on MDE for MRSs and different results. We mitigated this potential threat to validity by (1) carefully documenting the process we followed for building the classification framework, (2) having the data extraction process conducted by multiple researchers in collaboration, and (3) making the raw extracted data available for independent verification.

8 Related work

The interest on software engineering for robotic systems is clearly growing and contributions range from discussing the general trends in software engineering for robotics [104], over benefits of MDE for specific platforms [105], to developing test-benches for MDE approaches [106] to contributing MDE-based reuse processes [107] or specific software engineering techniques [108] to robotics. The works by Brugali et al. [14,21,109] review the use of Component-based Software Engineering (CBSE) and MDE in robotics. A state of the art of service robotic engineering might be found in [15]. The study provides a comprehensive picture of (1) the practices applied by robotics industrial and academic practitioners, including processes, paradigms, languages, tools, frameworks, and reuse practices, (2) the distinguishing characteristics of robotics software engineering, and (3) recurrent challenges usually faced, together with adopted solutions. One of the highlighted challenges of the robotic domain is the lack of reusable solutions for recurrent problems in application development. In practice, integration is typically performed in an ad-hoc way, while being complicated by the wide range of disciplines practitioners working to develop a full-working robotic system have.

Also, the body of literature on systematic studies on robotics in relation to programming and developing software for it is rich and ranges from mapping studies on software architectures for robotic systems [110] to systematic literature reviews on teaching robot programming to novices [111]. Pons et al. [112] carried out a, fairly shallow, systematic review on the application of existing software engineering techniques to robotics. Feitosa et al. [113] reported on a mapping study on embedded software engineering for mobile robots. While they were seeking potential application of a set of known software engineering techniques to robotics, in this paper, we report on how a specific branch of software engineering, MDE, is used for mobile robots. Others focus on other relevant aspects of robotics, such as safety for mobile robots [1], safety certification [114], and the state of DSLs for robotics [83].

The literature study on the state-of-the-art of DSLs in robotics [83] investigates MDE for robotics through the lens of the RoboCup@Work competition. The authors classify 41 publications and focus on specific robotics subdomains related, such as structure modelling, planning, manipulation, or coordination. As such, the study has a limited scope of nine venues specific to modelling in robotics and focuses on DSLs, i.e. does not query the selected databases for the use of UML in this context. In the study, the authors find an increasing number of DSL contributions to robotics starting in 2010, which is in line with our finding that the youngest UML profile in the field was introduced 15 years ago.

There also are research results on MDE for robotics in other studies. For instance, in a study on model-driven techniques in automation, Yang et al. [115], identify the OpenRTM robotics middleware [116] being used in this context. While being interesting, the survey does not investigate the platforms, automation techniques, or further MDE-related questions in the context of robotics.

A systematic mapping study on software architectures for robotic systems is presented by Ahmad et al. [117]. In this survey over 56 publications, the authors investigate architecture modelling techniques, applied modelling languages, architectural concerns, and application domain these architecture modelling techniques are applied to. The authors find that UML component diagrams are the most popular modelling language for robotics architectures but do neither investigate which automation support is used, nor which tools are employed.

A study on formal specification and verification of the behaviour of autonomous robotic systems and related tools analyses 63 publications found with Google Scholar [118]. The study does not focus on MDE but identifies the formalisms applied to the verification of robotic systems (e.g. temporal logic or state-transition descriptions) and presents the model checking tools applied to robotics.

Hence, to the best of our knowledge, a systematic study on the broad use of MDE for mobile robots has not been carried out before.

9 Conclusions

This paper presents the design, conduction, and results of a systematic mapping study on model-driven engineering techniques for mobile robotic systems. Specifically, by following well-established guidelines on systematic literature reviews and mapping studies [19,61,103], we identified an initial set of 2809 potentially relevant studies and then we performed a rigorous selection process, eventually leading to a final set of 97 primary studies. We analysed each of the 97 primary studies and extracted relevant data for each of them in order

to provide a map of MDE approaches for MRSs. The data extraction phase was supported by an incrementally-defined classification framework, which can also be used by future researchers either for putting an already-existing approach into context or to understand scientifically interesting contribution they may make with their own approaches.

The main results of the study are summarised in the following (and expanded throughout the paper):

- Researchers tend to publish in specialized scientific venues at the intersection of software engineering and robotics, even though many approaches are generally applicable.
- Researchers largely focus on solution proposals and contribute novel MDE methods and tools for mobile robotic systems.
- There is a strong scientific interest on single general-purpose terrestrial robots.
- The number of MDE approaches for multiple robots is growing.
- There is a strong need for better validating MDE approaches for MRSs in order to sharply identify their limits, scope, applicability, and characteristics.
- The used modelling languages are more domain-specific than generic (e.g. UML-based) with a focus on both structural and behavioural aspects of the modelled robotic systems.
- We observed a strong focus on code generation and model transformations over model-based analysis or the usage of models at runtime.
- Researchers tend to validate MDE approaches for MRSs by applying them to simple examples, often involving single terrestrial robots.
- Sound and rigorous empirical experiments are not yet widely carried out, potentially hampering the transferability of the proposed approaches into real industrial contexts.
- The most used language engineering tools are EMF, MontiCore, and Xtext. However, we observed very little MDE tools reuse for MRSs.

By presenting and discussing the above-mentioned results, this study provides a comprehensive and replicable map of the state of the art about MDE approaches for MRSs. The results of this study form a solid foundation for future research in the area and will (hopefully) lead to the development of new (and better) MDE methods and techniques for the mobile robotic systems of the future.

Funding Open access funding provided by Mälardalen University.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adap-

tation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

Appendix A: primary studies

ID	Title	Authors	Year
P1	Using Physical Quantities in Robot Software Models	Burgueño, L., Mayerhofer, T., Wimmer, M., Vallecillo, A. (2018
P2	SeMo: Service-Oriented and Model-Based Software Framework for Cooperating Robots	Hong, H., Jung, H., Park, K., Ha, S.	2018
P3	A Component-Based and Model-Driven Approach to Deal with Non-Functional Properties through Global QoS Metrics	Vicente-Chicote, C., Inglés-Romero, J. F., Martínez, J.	2018
P4	Modularization of Domain-Specific Languages for Extensible Component-Based Robotic Systems	Dennis Leroy Wigand, Arne Nordmann, Michael Goerlich, Sebastian Wrede	2017
P5	Managing the Functional Variability of Robotic Perception Systems	Davide Brugali, Nico Hochgeschwender	2017
P6	Executing Robot Task Models in Dynamic Environments	K Adam, A Butting, O Kautz, B Rumpe, A Wortmann	2017
P7	DRONA: A Framework for Safe Distributed Mobile Robotics	Desai, A., Saha, I., Yang, J., Qadeer, S., Seshia, S. A.	2017
P8	A Scenario-based MDE Process for Developing Reactive Systems: A Cleaning Robot Example.	Greenyer, J., Gritzner, D., Shi, J., Wete, E.	2017
P9	Model-based engineering of autonomous systems using ontologies and metamodels	Julita Bermejo-Alonso, Carlos Hernandez, Ricardo Sanz	2016
P10	Combining robotics component-based model-driven development with a model-based performance analysis	Alex Lotz, Arne Hamann, Ralph Lange, Christian Heinzemann, Jan Staschulat, Vincent Kesel, Dennis Stampfer, Matthias Lutz, Christian Schlegel	2016
P11	Adopting MDE for Specifying and Executing Civilian Missions of Mobile Multi-Robot Systems	F. Ciccozzi, D. Di Ruscio, I. Malavolta, P. Pelliccione	2016
P12	ROSMOD: a toolsuite for modeling, generating, deploying, and managing distributed real-time component-based software using ROS	Pranav Srinivas Kumar, William Emfinger, Gabor Karsai, Dexter Watkins, Benjamin Gasser, Amrutur Anilkumar	2016
P13	Model-Driven Development Architectures to Solve Complex Autonomous Robotics Problems	Alain Beaulieu, Sidney N. Givigi, Dany Ouellet, Jay Thor Turner	2016
P14	Model-driven engineering approach to design sensing and actuation subsystems	Fernando Silvano Goncalves, Leandro Buss Becker	2016
P15	Model-based development of a controller and simulator for a mobile robot.	M. K. Oplenskedal, P. Herrmann, J. O. Blech	2016

ID	Title	Authors	Year
P16	A Toolset to Address Variability in Mobile Robotics.	Tewfik Ziadi, Jean-Loup Farges, Serge Stinckwich, Mikal Ziane, Saadia Dhouib, Francois Marmoiton, Nicolas Morette, Cyril Novales, Selma Kchir, Bruno Patin	2016
P17	Applying MDA and OMG Robotic Specification for Developing Robotic Systems.	C Pons, G PÃ©rez, R Giandini, G Baum	2016
P18	Formalizing the specifications of a domain-specific language for authoring behaviour of personal service robots.	Chandan Datta, Elizabeth Broadbent, Bruce A MacDonald	2016
P19	Model-driven separation of concerns for service robotics	“Kai Adam, Arvid Butting, Robert Heim, Oliver Kautz, Bernhard Rumpe, Andreas Wortmann”	2016
P20	Model-based generation of a requirements monitor	Kneer F., Kamsties E.	2015
P21	Modeling robot and world interfaces for reusable tasks	Robert Heim, Pedram Mir Seyed Nazari, Jan Oliver Ringert, Bernhard Rumpe, Andreas Wortmann	2015
P22	A model-driven framework for distributed simulation of autonomous systems	P. Bocciarelli, A. D’Ambrosio, A. Giglio, E. Paglia	2015
P23	Language and Code Generator Composition for Model-Driven Engineering of Robotics Component & Connector Systems	Jan Oliver Ringert, Alexander Roth, Bernhard Rumpe, Andreas Wortmann	2015
P24	A model-driven engineering methodology for embedded system designs—HIPAO2	Doering, D.	2014
P25	A model-based design space exploration for embedded image processing in industrial applications	Lerm, R.; Doering, D.; Rech, R.H.A.; Pereira, C.E.; Rettberg, A.	2014
P26	A Modeling Framework for Software Architecture Specification and Validation	Gobillot, Nicolas, Charles Lesire, and David Doose	2014
P27	BRIDE-A toolchain for framework-independent development of industrial service robot applications	Bubeck, Alexander; Weisshardt, Florian; Verl, Alexander	2014
P28	Control and Scheduling Co-design for a Simulated Quadcopter Robot: A Model-Driven Approach	Matteo Morelli, Marco Natale	2014
P29	Dynamic State Charts: Composition and coordination of complex robot behaviour and reuse of action plots	Stampfer D., Schlegel C.	2014
P30	Extending the Kouretes Statechart Editor for Generic Agent behaviour Development	Papadimitriou, G. L., Spanoudakis, N. I., Lagoudakis, M. G.	2014
P31	How can model-driven development approaches improve the certification process for UAS?	Larrieu, N.	2014
P32	Model-driven analysis and design for software development of autonomous underwater vehicles	Ortiz F.J., Insaurralde C.C., Alonso D., Sanchez F., Petillot Y.R.	2014
P33	SafeRobots: A model-driven Framework for developing Robotic Systems	Ramaswamy, A.; Monsuez, B.; Tapus, A.	2014
P34	Towards Rule-Based Dynamic Safety Monitoring for Mobile Robots	Adam, Sorin and Larsen, Morten and Jensen, Kjeld and Schultz, Ulrik Pagh	2014
P35	A CASE tool for robot behaviour development	Topalidou-Kyniazopoulou A., Spanoudakis N.I., Lagoudakis M.G.	2013
P36	A hierarchical modeling method based on model-driven development in real-time control system design	Bi He; Bin Wang; Li Guo; Tongyao Yang; Xin Xiong	2013

ID	Title	Authors	Year
P37	A model-based approach to software deployment in robotics	Hochgeschwender, N.; Gherardi, L.; Shakhirmardanov, A.; Kraetzschmar, G.K.; Brugali, D.; Bruyninckx, H.	2013
P38	A Model-Driven Software Component Framework for Fractionated Spacecraft	Dubey, Abhishek; Gokhale, Aniruddha; Karsai, Gabor; Otte, William R.; Willemssen, Johnny	2013
P39	An MDE solution for the development of robotic device drivers	Conner Hansen, Eugene Syriani	2013
P40	An object-oriented analysis and design model to implement controllers for quadrotor uavs by specializing MDA's features with hybrid automata and real-time UML	Diem P.G., Hien N.V., Khanh N.P.	2013
P41	Dealing with run-time variability in service robotics: towards a DSL for non-functional properties	InglÄs-Romero, Juan Francisco; Lotz, Alex; Chicote, Cristina Vicente; Schlegel, Christian	2013
P42	Model-driven development of safe self-optimizing mechatronic systems with MechatronicUML	Giese H., Schafer W.	2013
P43	Model-driven safety assessment of robotic systems	Yakymets, N.; Dhoubi, S.; Jaber, H.; Lanusse, A.	2013
P44	Rapid application development of constrained-based task modelling and execution using domain specific languages	Vanthienen, Dominick; Klotzbuecher, M.; De Schutter, Joris; De Laet, Tinne; Bruyninckx, Herman	2013
P45	Towards Easy Robot Programming-Using DSLs, Code Generators and Software Product Lines.	Baumgartl, Johannes; Buchmann, Thomas; Henrich, Dominik; Westfechtel, Bernhard	2013
P46	The BRICS component model: a model-based development paradigm for complex robotics software systems	Herman Bruyninckx, Markus Klotzbuecher, Nico Hochgeschwender, Gerhard Kraetzschmar, Luca Gherardi, Davide Brugali	2013
P47	A visual modeling language for RDIS and ROS nodes using ATOM3	Paul Kilgo, Eugene Syriani, Monica Anderson	2012
P48	A domain specific language for modeling differential constraints of mobile robots	Guarnieri, Marco; Magri, Eros; Brugali, Davide; Gherardi, Luca	2012
P49	A Reuse-oriented Development Process for Component-based Robotic Systems	Brugali, Davide and Gherardi, Luca and Bizzi, A. and Luzzana, Andrea and Zakharov, Alexey	2012
P50	A software platform for fractionated spacecraft	Dubey, A.; Emfinger, W.; Gokhale, A.; Karsai, G.; Otte, W.R.; Parsons, J.; Szabo, C.; Coglio, A.; Smith, E.; Bose, P.	2012
P51	From Teleo-Reactive specifications to architectural components: A model-driven approach	Sanchez, P., Alonso, D., Morales, J. M., Navarro, P. J.	2012
P52	Model-driven behaviour specification for robotic teams	Alexandros Paraschos, Nikolaos I. Spanoudakis, Michail G. Lagoudakis	2012
P53	Model-driven engineering techniques for the development of multi-agent systems	Gascuena JM; Navarro E; Fernandez-Caballero A;	2012
P54	Robotml, a domain-specific language to design, simulate and deploy robotic applications	Dhoubi, Saadia; Kchir, Selma; Stinckwich, Serge; Ziadi, Tewfik; Ziane, Mikal	2012
P55	A Model-Based Approach for Reconciliation of Polychronous Execution Traces	Garces, K.; DeAntoni, J.; Mallet, F.	2011
P56	A modelling language for cooperative plans in highly dynamic domains	Skubch, Hendrik; Wagner, Michael; Reichle, Roland; Geihs, Kurt	2011
P57	Addressing systems verification of autonomous systems through Bi-directional model transformations: A systems model-driven architecture approach	Callow, G.; Kalawsky, R.; Watson, G.; Okuda, Y.	2011
P58	Applying software engineering techniques to the development of robotic systems	Pons, Claudia; AÄvalo, Gabriela Beatriz; Zabala, Gonzalo; Morain, Ricardo	2011

ID	Title	Authors	Year
P59	Control of swarms of autonomous robots using Model-Driven Development—A state-based approach	Ouellet, D.; Givigi, S.N.; Beaulieu, A.J.G.	2011
P60	Engineering Heterogeneous Robotics Systems: A Software-Architecture-Based Approach	Medvidovic, Nenad; Tajalli, Hossein; Garcia, Joshua; Krka, Ivo; Brun, Yuriy; Edwards, George	2011
P61	Improving Real-Time software design with component and aspect approaches	Hamouche, R.; Abdeddaim, Y.; Kocik, R.	2011
P62	Managing execution variants in task coordination by exploiting design-time models at run-time	Steck, Andreas; Schlegel, Christian	2011
P63	Model-driven engineering and run-time model-usage in service robotics	Andreas Steck, Alex Lotz, Christian Schlegel	2011
P64	Tiriba—a new approach of UAV based on model-driven development and multiprocessors	Branco, K.R.L.J.C.; Pelizzoni, J.M.; Oliveira Neris, L.; Trindade, O.; Osorio, F.S.; Wolf, D.F.	2011
P65	Towards the Automatic Generation of Self-Adaptive Robotics Software: An Experience Report	InglÃs-Romero, J.F.; Vicente-Chicote, C.; Morin, B.; Barais, O.	2011
P66	A platform-independent programming environment for robot control	Reckhaus, Michael; Hochgeschwender, Nico; Ploeger, Paul G.; Kraetzschmar, Gerhard K.	2010
P67	A systematic testing approach for autonomous mobile robots using domain-specific languages	Proetzsch M., Zimmermann F., Eschbach R., Kloos J., Berns K.	2010
P68	From UML/MARTE to RTDT: A model-driven-based method for scheduling analysis and HW/SW partitioning	Kacem, Y.H.; Mahfoudhi, A.; Tmar, H.; Abid, M.	2010
P69	Product line development using multiple domain specific languages in embedded systems	Tokumoto S.	2010
P70	Carpe noctem 2009	T. Amma, P. Baer, K. Baumgart, P. Burghardt, K. Geihs, J. Henze, S. Opfer, S. Niemczyk, R. Reichle, D. Saur, A. Scharf, J. Schreiber, M. Segatz, S. Seute, H. Skubch, S. Triller, M. Wagner, A. Witsch	2009
P71	Concern separation for adaptive QoS modeling in distributed real-time embedded systems	Gray J., Neema S., Zhang J., Lin Y., Bapty T., Gokhale A., Schmidt D.C.	2009
P72	Design of service robots	Iborra, A.; Caceres, D.; Ortiz, F.; Franco, J.; Palma, P.; Alvarez, B.	2009
P73	MDD-based CASE Tool for Modeling Heterogeneous Multi-jointed Robots	WooYeol Kim; Hyun Seung Son; Kim, R.Y.C.; Carlson, C.R.	2009
P74	Robotic software systems: From code-driven to model-driven designs	Schlegel, C.; Hassler, T.; Lotz, A.; Steck, A.	2009
P75	Avida-MDE: A digital evolution approach to generating models of adaptive software behaviour	Goldsby H.J., Cheng B.H.C.	2008

ID	Title	Authors	Year
P76	The SPICA development framework model-driven software development for autonomous mobile robots	Baer P.A., Reichle R., Geihs K.	2008
P77	A Domain-independent Middleware Framework for Context-aware Applications	Pinto, R. P., Cardozo, E., Coelho, P. R., Guimaraes, E. G.	2007
P78	Model-Driven Design of Reliable Robot Control Programs Using the jABC	Jorges, S.; Kubczak, C.; Pageau, F.; Margaria, T.	2007
P79	A Model-Driven Architecture Approach for Modeling, Specifying and Deploying Policies in Autonomous and Autonomic Systems	Pena, J.; Hinchey, M.G.; Sterritt, Roy; Ruiz-Cortes, A.; Resinas, M.	2006
P80	A model-driven approach for generating embedded robot navigation control software	Bina Shah, Rachael Dennison, Jeff Gray	2004

References

- Bozhinoski, D., Ruscio, D.D., Malavolta, I., Pelliccione, P., Crnkovic, I.: Safety for mobile robotic systems: a systematic mapping study from a software engineering perspective. *J. Syst. Softw.* **151**, 150–179 (2019)
- Self-driving electric bus propels Swiss town into the future. <https://edition.cnn.com/2018/06/27/sport/trapeze-self-driving-autonomous-electric-bus-switzerland-spt-intl> (2018). Accessed: 29 Nov 2018
- Meet the Cobots: Humans and Robots Together on the Factory Floor. <https://news.nationalgeographic.com/2016/05/financial-times-meet-the-cobots-humans-robots-factories/> (2018). Accessed: 27 Nov 2018
- Adam, K., Butting, A., Heim, R., Kautz, O., Pfeiffer, J., Rumpe, B., Wortmann, A.: Modeling Robotics Tasks for Better Separation of Concerns, Platform-Independence, and Reuse, ser. *Aachener Informatik-Berichte, Software Engineering*, Band 28. Shaker Verlag (2017)
- Friedenthal, S., Moore, A., Steiner, R.: A practical guide to SysML: the systems modeling language. Morgan Kaufmann, London (2014)
- Nordmann, A. Hochgeschwender, N., Wrede, S.: A survey on domain-specific languages in robotics. In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 195–206. Springer (2014)
- Malavolta, I., Lago, P., Muccini, H., Pelliccione, P., Tang, A.: What industry needs from architectural languages: a survey. *IEEE Trans. Softw. Eng.* **39**(6), 869–891 (2013)
- Stampfer, D., Lotz, A., Lutz, M., Schlegel, C.: The smartmdsd toolchain: an integrated mdsd workflow and integrated development environment (ide) for robotics software. *J. Softw. Eng. Robot. (JOSER)* **7**(08), 3–19 (2018)
- Bruyninx, H.: Open robot control software: the OROCOS project. In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, 2523–2528 (2001)
- Bozhinoski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P., Tivoli, M.: Flyaq: enabling non-expert users to specify and generate missions of autonomous multicopters. In: *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pp. 801–806. IEEE (2015)
- Garcia, S., Pelliccione, P., Menghi, C., Berger, T., Bures, T.: High-level mission specification for multiple robots. In: *12th ACM SIGPLAN International Conference on Software Language Engineering (SLE)*s (2019)
- Thomas, U., Hirzinger, G., Rumpe, B., Schulze, C., Wortmann, A.: A new skill based robot programming language using UML/P Statecharts. In: *2013 IEEE International Conference on Robotics and Automation*, pp. 461–466. IEEE (2013)
- Mühe, H., Angerer, A., Hoffmann, A., Reif, W.: On reverse-engineering the kuka robot language. In: *1st International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob-10)*, October 2010, Taipei, Taiwan, during the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10). [arXiv:1009.5004](https://arxiv.org/abs/1009.5004). (2010)
- Brugali, D.: Model-driven software engineering in robotics. *IEEE Robot. Automat. Mag.* **22**(3), 155–166 (2015)
- García, S., Strüder, D., Brugali, D., Berger, T., Pelliccione, P.: Robotics software engineering: a perspective from the service robotics domain. In: *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ser. *ESEC/FSE 2020*. New York, NY, USA: Association for Computing Machinery, pp. 593–604. <https://doi.org/10.1145/3368089.3409743> (2020)
- Sprinkle, J., Eklund, J.M., Gonzalez, H., Grøtli, E.I., Upcroft, B., Makarenko, A., Uther, W., Moser, M., Fitch, R., Durrant-Whyte, H., et al.: Model-based design: a report from the trenches of the darpa urban challenge. *Softw. Syst. Model.* **8**(4), 551–566 (2009)
- Dragomir, I., Iosti, S., Bozga, M., Bensalem, S.: Designing systems with detection and reconfiguration capabilities: a formal approach. In: Margaria, T., Steffen, B. (eds.) *Leveraging Applications of Formal Methods, Verification and Validation, Distributed Systems*, pp. 155–171. Springer, Cham (2018)
- Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M.: Systematic mapping studies in software engineering. In: *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, Series EASE'08*, pp. 68–77. British Computer Society, Swinton, (2008)
- Petersen, K., Vakkalanka, S., Kuzniarz, L.: Guidelines for conducting systematic mapping studies in software engineering: an update. *Inf. Softw. Technol.* **64**, 1–18 (2015)

20. France, R., Rumpe, B.: Model-driven development of complex software: a research roadmap. In: *Future of Software Engineering 2007* at ICSE, pp. 37–54 (2007)
21. Brugali, D., Agah, A., MacDonald, B., Nesnas, I.A.D., Smart, W.D.: *Trends in Robot Software Domain Engineering*, pp. 3–8. Springer, Berlin, Heidelberg (2007)
22. Kozaczynski, W., Booch, G.: Component-based software engineering. *IEEE Softw.* **15**(5), 34 (1998)
23. Brooks, A., Kaupp, T., Makarenko, S., Williams, A., Oreback, A.: Towards component-based robotics. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 163–168. IEEE (2005)
24. Makarenko, A., Brooks, A., Kaupp, T.: Orca: components for robotics. In: *International Conference on Intelligent Robots and Systems (IROS)*, pp. 163–168. Citeseer (2006)
25. Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y.: ROS: an open-source robot operating system. In: *ICRA Workshop on Open Source Software*, vol. 3, No. 3.2, p. 5. Kobe, Japan (2009)
26. Hu, G., Tay, W.P., Wen, Y.: Cloud robotics: architecture, challenges and applications. *IEEE Netw.* **26**(3), 21–28 (2012)
27. Turnbull, L., Samanta, B.: Cloud robotics: formation control of a multi robot system utilizing cloud infrastructure. In: *Proceedings of IEEE Southeastcon 2013*, pp. 1–4 (2012)
28. Mohanarajah, G., Hunziker, D., D'Andrea, R., Waibel, M.: Rapyuta: a cloud robotics platform. *IEEE Trans. Autom. Sci. Eng.* **12**(2), 481–493 (2014)
29. Toris, R., Kammerl, J., Lu, D.V., Lee, J., Jenkins, O.C., Osentoski, S., Wills, M., Chernova, S.: Robot web tools: efficient messaging for cloud robotics. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4530–4537. IEEE (2015)
30. SPARC, Robotics 2020 Multi-Annual Roadmap. <https://eu-robotics.net/sparc/upload/about/files/H2020-Robotics-Multi-Annual-Roadmap-ICT-2016.pdf> (2016)
31. Whittle, J., Hutchinson, J., Rouncefield, M.: The state of practice in model-driven engineering. *IEEE Softw.* **31**(3), 79–85 (2014)
32. Schmidt, D.C.: Guest Editors introduction: model-driven engineering. *Computer* **39**(2), 25–31 (2006)
33. Völter, M., Stahl, T., Bettin, J., Haase, A., Helsen, S., Czarnecki, K.: *Model-Driven Software Development: Technology, Engineering, Management*. Wiley Software Patterns Series, Wiley, New York (2013)
34. Kühne, T.: What is a Model? In: *Language Engineering for Model-Driven Software Development*, number 04101 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), pp. 1–10. Schloss Dagstuhl (2005)
35. Sprinkle, J., Rumpe, B., Vangheluwe, H., Karsai, G.: Meta-modelling: State of the art and research challenges. In: *Model-Based Engineering of Embedded Real-Time Systems*, pp. 57–76. Springer (2010)
36. Grönniger, H., Krahn, H., Rumpe, B., Schindler, M., Völkel, S.: Monticore: a framework for the development of textual domain specific languages. In: *30th International Conference on Software Engineering (ICSE 2008)*, Leipzig, Germany, May 10–18, 2008, Companion Volume, pp. 925–926
37. Heldal, R., Pelliccione, P., Eliasson, U., Lantz, J., Derehag, J., Whittle, J.: Descriptive vs prescriptive models in industry. In: *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, ser. MODELS '16, pp. 216–226. ACM, New York, (2016)
38. Kuhn, A., Murphy, G.C., Thompson, C.A.: An exploratory study of forces and frictions affecting large-scale model-driven development. In: *Proceedings of MODELS'12*, pp. 352–367. Springer (2012)
39. Aranda, J., Damian, D., Borici, A.: Transition to model-driven engineering—what is revolutionary, what remains the same? In: *Proceedings of MODELS'12*, pp. 692–708. Springer (2012)
40. Baker, P., Loh, S., Weil, F.: Model-driven engineering in a large industrial context—Motorola Case Study. In: *Proceedings of MoDELS'05*, pp. 476–491. Springer Berlin, Heidelberg (2005)
41. Liebel, G., Marko, N., Tichy, M., Leitner, A., Hansson, J.: Assessing the state-of-practice of model-based engineering in the embedded systems domain. In: *Proceedings of MODELS'14*, 2014., pp. 166–182 (2014)
42. Vallecillo, A.: On the industrial adoption of model driven engineering. Is your company ready for MDE? *J. Inf. Syst. Softw. Eng. Big Compan. (IJISEBC)* **1**(1), 52–68 (2014)
43. Gorschek, T., Tempero, E., Angelis, L.: On the use of software design models in software development practice: an empirical investigation. *J. Syst. Softw.* **95**, 176–193 (2014)
44. Petre, M.: *UML in practice*. In: *Proceedings of ICSE '13*, pp. 722–731. IEEE Press (2013)
45. Cicciozzi, F., Crnkovic, I., Di Ruscio, D., Malavolta, I., Pelliccione, P., Spalazzese, R.: Model-driven engineering for mission-critical iot systems. *IEEE Softw.* **34**(1), 46–53 (2017)
46. Wortmann, A., Barais, O., Combemale, B., Wimmer, M.: Modeling languages in industry 4.0: an extended systematic mapping study. In: *Software and Systems Modeling*, pp. 1–28 (2019)
47. Hutchinson, J., Rouncefield, M., Whittle, J.: Model-driven engineering practices in industry. In: *Proceedings of the 33rd International Conference on Software Engineering (ICSE)*
48. Bruel, J.-M., Combemale, B., Ober, I., Raynal, H.: MDE in practice for computational science. In: *International Conference on Computational Science (ICCS 2015)*. Reykjavik, Iceland (2015)
49. Schlegel, C., Hassler, T., Lotz, A., Steck, A.: Robotic software systems: from code-driven to model-driven designs. In: *Advanced Robotics, 2009. ICAR 2009. International Conference on* (2009)
50. Steck, A., Lotz, A., Schlegel, C.: Model-driven engineering and run-time model-usage in service robotics. In: *Proceedings of the 10th ACM International Conference on Generative Programming and Component Engineering Ser. GPCE11*, pp. 73–82 (2011)
51. Bruyninckx, H., Klotzbücher, M., Hochgeschwender, N., Kraetzschmar, G., Gherardi, L., Brugali, D.: The BRICS component model: a model-based development paradigm for complex robotics software systems. In: *Proceedings of the 28th Annual ACM Symposium on Applied Computing*, ser. SAC '13, pp. 1758–1764. ACM, New York (2013)
52. Heim, R., Nazari, P.M.S., Ringert, J.O., Rumpe, B., Wortmann, A.: Modeling robot and world interfaces for reusable tasks. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1793–1798. IEEE (2015)
53. Cicciozzi, F., Ruscio, D.D., Malavolta, I., Pelliccione, P.: Adopting mde for specifying and executing civilian missions of mobile multi-robot systems. *J. IEEE Access* (2016)
54. Menghi, C., Tsigkanos, C., Berger, T., Pelliccione, P.: PsALM: specification of dependable robotic missions. In: *International Conference on Software Engineering (ICSE): Companion Proceedings* (2019)
55. Menghi, C., Tsigkanos, C., Pelliccione, P., Ghezzi, C., Berger, T.: Specification patterns for robotic missions. *Trans. Softw. Eng. (TSE)* <https://doi.org/10.1109/TSE.2019.2945329>
56. Akiki, P.A., Akiki, P.A., Bandara, A.K., Yu, Y.: EUD-MARS: end-user development of model-driven adaptive robotics software systems. *Sci. Comput. Program.* **200**, 102534 (2020)
57. Trower, J., Gray, J.: Blockly language creation and applications: Visual programming for media computation and bluetooth robotics control. In: *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pp. 5–5 (2015)
58. Weintrop, D., Afzal, A., Salac, J., Francis, P., Li, B., Shepherd, D.C., Franklin, D.: Evaluating coblox: a comparative study of

- robotics programming environments for adult novices. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–12 (2018)
59. Keele, S.: Guidelines for performing systematic literature reviews in software engineering. Technical report, EBSE Technical Report EBSE-2007-01. Technical report (2007)
60. Wieringa, R., Maiden, N., Mead, N., Rolland, C.: Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requir. Eng.* **11**(1), 102–107 (2006)
61. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* **55**(12), 2049–2075 (2013)
62. Ghezzi, C., Jazayeri, M., Mandrioli, D.: *Fundamentals of Software Engineering*, 2nd edn. Prentice Hall PTR, New York (2002)
63. Soley, R., et al.: Model driven architecture. *OMG White Pap.* **308**(308), 5 (2000)
64. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering*. Ser Computer Science, Springer, London (2012)
65. Zhang, H., Babar, M.A.: Systematic reviews in software engineering: an empirical investigation. *Inf. Softw. Technol.* **55**(7), 1341–1354 (2013)
66. Horsley, T., Dingwall, O., Sampson, M.: Checking reference lists to find additional studies for systematic reviews. *Cochrane Datab. Syst. Rev.* no. 8 (2011)
67. Wohlin, C.: Guidelines for snowballing in systematic literature studies and a replication in software engineering. In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, Ser. EASE '14, pp. 38:1–38:10. ACM, New York (2014)
68. Charnaz, K., Belgrave, L.L.: Grounded theory. In: *The Blackwell encyclopedia of sociology* (2007)
69. Cruzes, D.S., Dyba, T.: Recommended steps for thematic synthesis in software engineering. In: *International Symposium on Empirical Software Engineering and Measurement*, vol. 2011, pp. 275–284 (2011)
70. Franzosi, R.: *Quantitative Narrative Analysis*, vol. 162. Sage, London (2010)
71. Rodgers, M., Sowden, A., Petticrew, M., Arai, L., Roberts, H., Britten, N., Popay, J.: Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function. *Evaluation* **15**(1), 49–73 (2009)
72. Engström, E., Runeson, P.: Software product line testing—a systematic mapping study. *Inf. Softw. Technol.* **53**(1), 2–13 (2011)
73. Mehmood, A., Jawawi, D.N.: Aspect-oriented model-driven code generation: a systematic mapping study. *Inf. Softw. Technol.* **55**(2), 395–411 (2013). Special Section: Component-Based Software Engineering (CBSE) (2011)
74. Petersen, K.: Measuring and predicting software productivity: a systematic map and review. *Inf. Softw. Technol.*, **53**(4), 317–343 (2011). Special section: Software Engineering track of the 24th Annual Symposium on Applied Computing Software Engineering track of the 24th Annual Symposium on Applied Computing
75. Francesco, P.D., Lago, P., Malavolta, I.: Architecting with microservices: a systematic mapping study. *J. Syst. Softw.* **150**, 77–97 (2019)
76. Bell, R.: Introduction to IEC 61508. In: *Proceedings of the 10th Australian Workshop on Safety Critical Systems and Software*—Volume 55, pp. 3–12. Australian Computer Society, Inc. (2006)
77. Echeverria, G., Lemaignan, S., Degroote, A., Lacroix, S., Karg, M., Koch, P., Lesire, C., Stinckwich, S.: Simulating complex robotic scenarios with MORSE. In: *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, pp. 197–208. Springer (2012)
78. Andrews, T., Qadeer, S., Rajamani, S.K., Rehof, J., Xie, Y.: Zing: a model checker for concurrent software. In: *International Conference on Computer Aided Verification*, pp. 484–487. Springer (2004)
79. Selic, B., Limited, O.: Real-time object-oriented modeling (ROOM). In: *Proceedings of the 2Nd IEEE Real-Time Technology and Applications Symposium (RTAS '96)*, Ser. RTAS '96, pp. 214. IEEE Computer Society, Washington, DC (1996)
80. Feiler, P.H., Gluch, D.P., Hudak, J.J.: *The architecture analysis and design language (AADL): an introduction*. Carnegie-Mellon Univ Pittsburgh PA Software Engineering Inst, Technical Report (2006)
81. Weikens, T.: *Systems engineering with SysML/UML: modeling, analysis, design*. Elsevier, London (2011)
82. Dennis, S., Alex, L., Matthias, L., Christian, S.: The SmartMDS toolchain: an integrated MDS workflow and integrated development environment (IDE) for robotics software (2016)
83. Nordmann, A., Hochgeschwender, N., Wigand, D., Wrede, S.: A survey on domain-specific modeling and languages in robotics. *J. Softw. Eng. Robot.* **7**, 75–99 (2016)
84. Ivarsson, M., Gorschek, T.: A method for evaluating rigor and industrial relevance of technology evaluations. *Empir. Softw. Eng.* **16**(3), 365–395 (2011)
85. Runeson, P., Host, M., Rainer, A., Regnell, B.: *Case Study Research in Software Engineering: Guidelines and Examples*. Wiley, London (2012)
86. Kraetzschmar, G.K., Hochgeschwender, N., Nowak, W., Hegger, F., Schneider, S., Dwiputra, R., Berghofer, J., Bischoff, R.: Robocup@ work: competing for the factory of the future. In: *Robot Soccer World Cup*, pp. 171–182. Springer, Berlin (2014)
87. Boedecker, J., Asada, M.: Simspark-concepts and application in the robocup 3d soccer simulation league. *Auton. Robot.* **174**, 181 (2008)
88. Gerkey, B., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: *Proceedings of the 11th International Conference on Advanced Robotics*, vol. 1, pp. 317–323 (2003)
89. Berndt, J.: JSBSim: an open source flight dynamics model in C++. In: *AIAA Modeling and Simulation Technologies Conference and Exhibit*, p. 4923 (2004)
90. Steinberg, D., Budinsky, F., Merks, E., Paternostro, M.: *EMF: eclipse modeling framework*. Pearson Education, New York (2008)
91. Bettini, L.: *Implementing domain-specific languages with Xtext and Xtend*. Packt Publishing Ltd, London (2016)
92. Hölldobler, K., Rumpe, B.: *MontiCore 5 Language Workbench Edition 2017*, ser. Aachener Informatik-Berichte, Software Engineering, Band 32. Shaker (2017)
93. Datta, C., Jayawardena, C., Kuo, I.H., MacDonald, B.A.: RoboStudio: a visual programming environment for rapid authoring and customization of complex services on a personal service robot. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2352–2357. IEEE (2012)
94. Hasanain, W., Labiche, Y., Gheorghe, S.: Automated state-based online testing real-time embedded software with RTEdge. In: *2015 3rd International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pp. 294–302. IEEE (2015)
95. Nordmann, A., Hochgeschwender, N., Wigand, D.L., Wrede, S.: A survey on domain-specific modeling and languages in robotics. *J. Softw. Eng. Robot. (JOSER)* **7**(1), 75–99 (2016)
96. Stol, K.-J., Fitzgerald, B.: The ABC of software engineering research. *ACM Trans. Softw. Eng. Methodol.* **27**(3), 11:1–11:51 (2018)
97. Penzenstadler, B., Raturi, A., Richardson, D., Calero, C., Femmer, H., Franch, X.: Systematic mapping study on software engi-

- neering for sustainability (SE4S). In: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, p. 14. ACM (2014)
98. Engström, E., Runeson, P.: Software product line testing-a systematic mapping study. *Inf. Softw. Technol.* **53**(1), 2–13 (2011)
 99. Neto, P.A.M.S., Carmo Machado, I., McGregor, J.D., De Almeida, E.S., de Lemos Meira, S.R.: A systematic mapping study of software product lines testing. *Inf. Softw. Technol.* **53**(5), 407–423 (2011)
 100. Denil, J., Klikovits, S., Mosterman, P.J., Vallecillo, A., Vangheluwe, H.: The experiment model and validity frame in M&S. In: Proceedings of the Symposium on Theory of Modeling and Simulation, p. 10. Society for Computer Simulation International (2017)
 101. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. *J. Syst. Softw.* **80**(4), 571–583 (2007)
 102. Dyba, T., Dingsoyr, T., Hanssen, G.: Applying systematic reviews to diverse study types: an experience report. In: Empirical Software Engineering and Measurement: ESEM 2007 vol. 2007, pp. 225–234 (2007)
 103. Kitchenham, B.A., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Keele University and University of Durham, Technical Report EBSE-2007-01 (2007)
 104. Brugali, D., Prassler, E.: Software engineering for robotics [From the Guest Editors]. *IEEE Robot. Autom. Mag.* **16**(1), 9–15 (2009)
 105. Blanc, X., Delatour, J., Ziadi, T.: Benefits of the MDE approach for the development of embedded and robotic systems. In: Proceedings of the 2nd National Workshop on Control Architectures of Robots: From Models to Execution on Distributed Control Architectures, CAR (2007)
 106. Morelli, M., Moro, F., Rizano, T., Fontanelli, D., Palopoli, L., Natale, M.D.: A robotic vehicle testbench for the application of MBD-MDE development technologies. In: 2013 IEEE 18th Conference on Emerging Technologies and Factory Automation (ETFA), pp. 1–4. IEEE (2013)
 107. Brugali, D., Gherardi, L., Biziak, A., Luzzana, A., Zakharov, A.: A reuse-oriented development process for component-based robotic systems. In: International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pp. 361–374. Springer (2012)
 108. Ciccozzi, F., Crnkovic, I., Di Ruscio, D., Malavolta, I., Pelliccione, P., Spalazzese, R.: Model-driven engineering for mission-critical iot systems. *IEEE Softw.* **34**(1), 46–53 (2017)
 109. Brugali, D.: From the editor-in-chief: a new research community, a new journal. *J. Softw. Eng. Robot.* **1**, 1–2 (2010)
 110. Ahmad, A., Babar, M.A.: Software architectures for robotic systems: a systematic mapping study. *J. Syst. Softw.* **122**, 16–39 (2016)
 111. Major, L., Kyriacou, T., Brereton, O.P.: Systematic literature review: teaching novices programming using robots. *IET Softw.* **6**(6), 502–513 (2012)
 112. Pons, C., Giandini, R., Arévalo, G.: A systematic review of applying modern software engineering techniques to developing robotic systems. *Ingeniería e Investigación* **32**(1), 58–63 (2012)
 113. Feitosa, D., Felizardo, K.R., de Oliveira, L.B.R., Wolf, D.F., Nakagawa, E.Y.: Software engineering in the embedded software and mobile robot software development: a systematic mapping, pp. 738–741. In: SEKE (2010)
 114. Ingbergsson, J.T.M., Schultz, U.P., Kuhrmann, M.: On the use of safety certification practices in autonomous field robot software development: a systematic mapping study. In: International Conference on Product-Focused Software Process Improvement, pp. 335–352. Springer (2015)
 115. Yang, C.-H., Vyatkin, V., Pang, C.: Model-driven development of control software for distributed automation: a survey and an approach. *IEEE Trans. Syst. Man Cybern. Syst.* **44**(3), 292–305 (2013)
 116. Ando, N., Suehiro, T., Kotoku, T.: A software platform for component based rt-system development: Openrtm-aist. In: International Conference on Simulation, Modeling, and Programming for Autonomous Robots, pp. 87–98. Springer (2008)
 117. Ahmad, A., Ali Babar, M.: Software architectures for robotic systems: a systematic mapping study. *J. Syst. Softw.* **122**, 16–39 (2016)
 118. Luckcuck, M., Farrell, M., Dennis, L.A., Dixon, C., Fisher, M.: Formal specification and verification of autonomous robotic systems: a survey. *ACM Comput. Surv.* (2019). <https://doi.org/10.1145/3342355>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Giuseppina Lucia Casalaro is holding a Master's Degree in Computer Science from Università degli Studi di L'Aquila and a Master of Science in Computer Science with Specialization in Software Engineering from Mälardalen University in Sweden in 2015. Giuseppina started her career as CX Consultant, and now she works as Global IT Call Center Sr. Specialist in the world's largest hearing aid company, she is managing all the processes involving the final customer and responsible for the identification of the best solutions in compliance with market standards. In recent years she has been involved, together with her colleagues, in research in the field of MDE for MRSS.



Giulio Cattivera is a senior Software Engineer who deals with research and development of software systems as a specialized consultant. He works especially in the defense, banking and embedded systems industries. He has been a technical consultant for large leading companies such as Leonardo, Eurocontrol and Poste Italiane, but he also deals with technical managing of start-ups in the creation of digital products. In his work he deals with technological analysis, designing of software architectures and following all stages of development and verification. In 2015 he received a master's degree in Computer Science with a specialization in Software Engineering from the University of Aquila (Italy) and Mälardalen University (Sweden). In 2019 he obtained a postgraduate specialization degree in Sound Engineering at the University of Rome Tor Vergata.



Federico Ciccozzi is an Associate Professor at Mälardalen University (Sweden). His research specializes in: definition of DSMLs, model transformations, system properties preservation, multi-paradigm modelling, model versioning, combination of MDE and CBSE for complex systems, blended modelling, language and compiler engineering. Federico has organized over 40 conference tracks, sessions, workshops and journal special issues. He has been program committee member of over

40 scientific events in the last year. He is associate editor for IET Software, and 6 times guest editor of SoSyM and JISA. He has (co-)authored over 100 peer-reviewed publications. More info at: http://www.es.mdh.se/staff/266-Federico_Ciccozzi



Ivano Malavolta is Assistant professor at the Computer Science Department and Director of the Network Institute at the Vrije Universiteit Amsterdam (The Netherlands). His research focuses on data-driven software engineering, with a special emphasis on software architecture, mobile software development, robotics software. He applies empirical methods to evaluate methods and techniques in the field of software engineering. He authored several scientific articles in international journals and

peer-reviewed international conferences proceedings. He is a program committee member and reviewer of international conferences and journals in the software engineering field. He received a PhD in computer science from the University of L'Aquila in 2012. He is a member of ACM, IEEE, VERSEN, Amsterdam Young Academy, and Amsterdam Data Science.



Andreas Wortmann is an associate professor at Stuttgart University, where he is a member of the Institute for Control Engineering of Machine Tools and Manufacturing Units (ISW). He received his Ph.D. from RWTH Aachen University in 2016 where he led a group on model-driven systems engineering. He is a member of the IEEE Technical Committee on Software Engineering for Robotics and Automation and a board member of the European Association for Programming Languages and

Systems. His research interests include model-driven engineering, systems engineering, software language engineering, and robotics. He has co-authored over 80 international conference and journal publications. For more information, please visit <https://awortmann.github.io/>.



Patrizio Pelliccione is a Full Professor at Gran Sasso Science Institute (GSSI) and an Associate Professor at the Department of Computer Science and Engineering at Chalmers University of Gothenburg. He got his Ph.D. in 2005 at the University of L'Aquila (Italy) and from February 1, 2014, he is Docent in Software Engineering, a title given by the University of Gothenburg. His research topics are mainly in software engineering, software architecture modeling and verification, autonomous

systems, and formal methods. He has co-authored more than 150 publications in journals and international conferences and workshops on these topics. He has been on the organization and program committees for several top conferences and he is a reviewer for top journals in the software engineering domain. He is very active in European and National projects. In his research activity, he has collaborated with several companies. More information is available at <http://www.patriziopelliccione.com>.